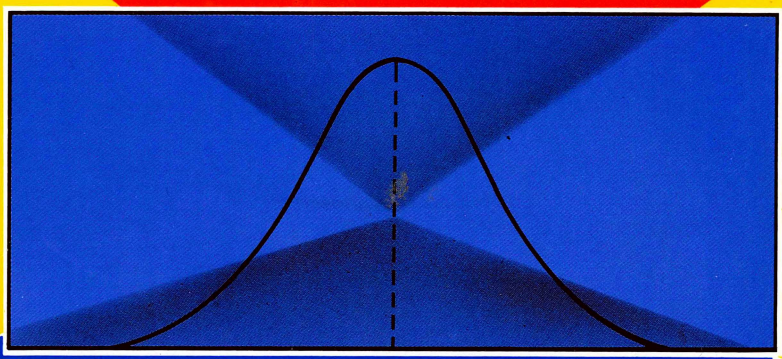
OSF
COMPUTERS, INC.
$24.95

THE GRADEBOOK SYSTEM



APPLE II/IIe

JOHN BLANKENSHIP

BOOKWARE™

OSF COMPUTERS, INC.
$ 24.95
SAPL0190
GRADEBOOK

THE GRADEBOOK SYSTEM

It is no longer necessary to spend a fortune for quality software. Prentice-Hall, Inc. and the author take great pride in pioneering a new low-cost method for distributing software. Any school or university that purchases one book/diskette combination of *The Gradebook System* is authorized to make copies of the diskette for every faculty member for whom the documentation (book alone) has been purchased. Your adherence to this policy will help to ensure the future availability of low-cost software packages.

For additional information or to obtain a copy of the software to accompany *The Gradebook System*, please send a check or money order for $20.00 to:

Prentice-Hall, Inc.
Attention: Ryan Colby
College Operations
Englewood Cliffs, New Jersey 07632

In addition, if your diskette is defective or damaged in transit, return it directly to Prentice-Hall at the address above for a no-charge replacement within 90 days of the date of purchase. Mail the defective diskette together with your name and address.

Limits of Liability and Disclaimer of Warranty:

The author and publisher of this book have used their best efforts in preparing this book and software. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs of the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

THE GRADEBOOK SYSTEM

(APPLE II/IIe)

JOHN BLANKENSHIP

Senior Professor
DeVry Institute of Technology

PRENTICE-HALL, INC.
Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

Blankenship, John (date)
The gradebook system.

1. Computer managed instruction. 2. Grading and marking (Students)—Computer programs. I. Title.
LB1028.46.B58 1984 371.2'72'02854 84-4901
ISBN 0-13-362542-7 (set)
ISBN 0-13-362526-5(hard)
ISBN 0-13-362534-6 (disk)

Editorial/production supervision: Nancy Milnamow
Cover design: George Cornell
Manufacturing buyer: Gordon Osbourne

Apple is a registered trademark of Apple Computer, Inc.

© 1984 by Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book may be
reproduced, in any form or by any means,
without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-362542-7 {BOOK AND DISK}
ISBN 0-13-362526-5 {BOOK}
ISBN 0-13-362534-6 {DISK}

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

CONTENTS

Chapter 1	AN INTRODUCTION	1
Chapter 2	ENTERING NAMES	4
Chapter 3	DISK OPERATIONS	6
Chapter 4	ENTERING GRADES	9
Chapter 5	EXAMINING GRADES	13
Chapter 6	MAKING MODIFICATIONS	16
Chapter 7	ANALYZING AND ADJUSTING GRADES	18
Chapter 8	USING THE CARD READER	25
Chapter 9	MODULE DESCRIPTIONS	32
Chapter 10	MODIFYING DATA FILES	43
Chapter 11	A FINAL WORD	45
Appendix A	GRADEBOOK	49
Appendix B	FLEXIBLE SORT	65

THE GRADEBOOK SYSTEM

Chapter 1

AN INTRODUCTION

The record-keeping responsibility of a teacher usually is very tedious and time consuming, but it does not have to be. With my program and the appropriate equipment you can grade the tests for an entire class in only a few minutes. And in that few minutes you will get more than grades.

Naturally, you will receive an analysis of each test, such as the high and low grades, as well as the class average. You can print a histogram to display the grade distributions. Perhaps even more important though, you can get an analysis of the questions. This analysis will tell you how many people missed each question and summarize what answers they chose. Imagine how much help such information could be when going over a test with your class or preparing a new one.

Grading tests is only the beginning. Don't expect to have to enter grades from the keyboard. If you use the recommended card reader (see Chapter 8) to grade the tests, the scores will automatically be entered into the computer. Because of the increased efficiency of the faculty, I highly recommend that school systems provide a card reader. If such a purchase is impossible, you need not worry. My program also provides for easy entry of grades from the keyboard.

Once the computer has the scores, obtained from either the card reader or the keyboard, the data can be manipulated in most any way you desire. In this area, I think that my dozen or so years of experience

as a teacher have really paid off. I know what teachers want a gradebook program to do.

To begin with, my program will allow you to curve the grades. You may curve them manually using a linear function, or you might choose one of the two automatic modes and let the computer handle the statistics for you.

In addition to curving, you can drop the lowest grade (or grades) for each student. You can even replace the lowest grade with some other grade, such as that of the final exam. When doing either of the above, you don't have to consider all the tests. You can choose just homework or just quizzes. For that matter, you can even use some of each.

It is this flexiblity that makes the program so usable. Unlike many gradebook programs I have seen, you won't have to specify the number of tests and quizzes you plan to give in advance. You will be able to change your mind at any time, about anything.

You will be able to get reports, such as which students are making A's or which ones are failing. The students' grades are valid at any time because a running average is used. It is not necessary to wait until the end of the term to determine how things are going.

The student data is easily edited. You may drop students, excuse tests, inquire about a specific student . . . the list goes on and on. And if all this isn't enough, I'm even going to show you how to make custom modifications so that your Gradebook can be exactly the way you want it.

I have tantalized you enough. The best way to learn to use The Gradebook System is to try it. Before you start, however, make a back-up copy of the disk that came with this book. It can be copied using any standard copy program. Now you are ready. Just boot the disk (or RUN HELLO) and you will see the following menu.

```
N-NAMES (ENTER)
T-TESTS QUIZZES HOMEWORK (ENTER)
M-MODIFY DATA
W-WEIGHTS (MODIFY)
A-AVERAGES (LIST)
G-GRADES (LIST)
F-FIND ONE STUDENTS GRADES
B-BETWEEN TWO GRADES (LIST)
S-SAVE GRADES TO DISK
L-LOAD GRADES FROM DISK
C-CATALOG
K-CURVE GRADES
```

```
R-REPLACE LOWEST GRADE
O-ORDER GRADES FOR DROPPING
H-HISTOGRAM (STATISTICAL ANALYSIS)
P-PRINTER (ON/OFF)
E-END PROGRAM
SELECT LETTER OF YOUR CHOICE
```

In order to keep the number of key strokes to a minimum, you don't have to press RETURN for any input that can be accomplished with a single key. Consequently, you may select any of the menu items by pressing just the corresponding letter. The above options will be covered in detail throughout the next seven chapters in order to guide you through all of the capabilities of the Gradebook System. After you have learned to use the program as it was written, I will devote some space to showing you how to customize it for your particular needs.

Chapter 2

ENTERING NAMES

Before grades can be entered or analyzed, the names of the students must be entered. To enter student names into the Gradebook Program, choose the N option from the menu. Since there are no students presently in the Gradebook, the computer will print:

```
YOU MAY NOW ENTER NEW NAMES
ENTER EACH NAME–JUST RTN WHEN DONE
```

The program will number each of the students as you enter them. Let's try it out. Use these exact names if you don't mind, since all the examples throughout the book will refer to them. Type in the following names.

```
SMITH JOHN
WHITE WENDY
JONES TOM
```

The computer should now be expecting you to enter the fourth name. Without typing any letters this time, just hit the RETURN key. The computer will ask:

```
DO YOU WISH TO SORT THESE NAMES (Y/N)
```

Press N for no and you should return to the menu.

Let's again choose the N-NAMES option. This time we get a different response. The computer prints:

```
THERE ARE NOW 3 STUDENTS ON FILE
DO YOU WISH TO ENTER A NEW CLASS OR ADD
TO THE PRESENT ONE (E/A)
```

Press A and we will add some names to those we already have. Start by adding:

```
SAM SMART
```

After hitting RETURN you suddenly realize you should have put the name in last name first. Don't despair. This can be easily corrected. For the next name, enter just the character <. When you hit RETURN the previous number will be repeated and you may re-enter Sam's name as

```
SMART SAM
```

You may use the < key to back up as many times as you wish. When you get to where you are going, though, you will have to type in all the names from that point. This means that if you have entered 50 names and you realize that the third one is misspelled, you should not back up and fix it. We will see later how to correct such mistakes.

Let's enter one more name to make a total of five. Type in the following:

```
DOE JANE
```

For this sixth name, just hit RETURN to indicate that you have finished entering names. This time, ask the program to sort the names by responding with a Y. After a short pause, you should see the menu. This is a good place to point out that the alphabetizing option will not be available to you once some grades have been entered. You will be able to add or delete names, but you cannot sort them (the reason for this concerns the card reader and will be explained in Chapter 8). Chapter 10 will show you how to side step this limitation if you just can't live without the sort.

Now that the names of the students have been entered, the next chapter will show you how to save them to the disk.

Chapter 3

DISK OPERATIONS

Even though you have not entered any grades yet, I will show you how to save files on a diskette. Select this option by typing a S from the menu. The computer will respond with a flashing warning to remind you that a data file on the disk will be replaced when you use this command. You must be careful here or you could lose a lot of data.

For example, suppose you have two classes and you have saved their grades under the names MATH and HISTORY. Also assume that the class you are about to save is the MATH class. If you tell the computer to use the name HISTORY when you save the data, you will end up with two sets of MATH grades although one will be named HISTORY. In just a moment I will show you what steps, in addition to the warning, I have taken to prevent this from occurring.

The computer will also print:

```
READY TO SAVE DATA TO DISK
WHAT FILE NAME
```

You should respond by typing in the name of the class. It must start with a letter but can contain both letters and numbers. For this example type in:

```
MATH 101
```

When you press RETURN, the disk drive will come on for a few seconds and then the menu will appear. If you would like to look at the names of the files on the disk, press C for the CATALOG option. It will respond by asking:

```
DRIVE 1 OR 2
```

In this case, press 1. You will see the programs that come on the Gradebook diskette and a text file named MATH 101. Hit any key to get back to the menu.

The ability to select either drive 1 or drive 2 with the catalog option provides some added flexibility. Some teachers will prefer not to save their class data on the program diskette. You may use any normally-formatted diskette to save your data.

If you have two drives, you may place the program in drive 1 for booting and the data diskette in drive 2. If you plan to do a lot of backups, you may wish to use both drives for data. The program diskette is not needed once the Gradebook Program is in memory.

The LOAD and SAVE options for the Gradebook Program always use the last drive that was cataloged. When ever you desire to access a file on the opposite drive, just CATALOG that drive before you perform the LOAD or SAVE.

Now that we are back at the menu, let's choose the S option again. Since a file name has been used before, the program will now ask if you wish to use it again. If you do, just press Y and the file will be saved under that name. Pressing N will cause the computer to ask what name you wish to use. If you have changed your mind and do not want to save the data at all, just hit RETURN to get back to the menu. Throughout the program, this return option will almost always abort the present action and return you to the menu.

One additional safety factor is built into the disk operations. It came about because of the following incident. I was preparing final grades early one Saturday morning. I had booted the disk and was trying to load a class file. Unfortunately, I accidentally selected the SAVE option and typed in the name. When I hit RETURN, a null or empty file was saved on top of my data. I did have a backup copy, but it reminded me how easily data could be lost.

I quickly modified the program so that it will not allow the save command to operate if there are no students in the memory. This change, coupled with the flashing warning and the ability to use the previous file name, should prevent most disasters.

Notice that, since no grades have been entered, the file was saved contains only the student names. This ability to save only the names can be very beneficial if you have more than one class with the same, or

nearly the same, people. This occurs very often for me as I usually have both a lecture and a lab with the same students.

I save the student names under a file such as PHYSICS 101 NAMES. It can be loaded and used to start both the lab and the lecture files. Naturally, once I have entered any grades, I save the files under such names as PHYSICS 101 LECTURE and PHYSICS 101 LAB. The original file name can then be deleted by using Apple DOS, if so desired.

The L (LOAD) command is very similar to the S (SAVE) command except that it loads the file whose name is asked for rather than saving it. You will always have to supply the file name during loads. It is very unlikely that you would wish to LOAD a file that you already have in memory, so the option of using the previous file name is not offered.

Now that we can SAVE and LOAD the names we have typed in, let's move on to the next chapter and see how grades can be entered.

Chapter 4

ENTERING GRADES

From the main menu, you may begin entering grades by pressing the T key. If you have not entered any names, you will be instructed to do so and any key will return you to the menu. In this case, I will assume that you are using the five names from Chapter 2. The computer will respond with the following statements.

```
THERE ARE NOW 0 TESTS
WHAT SET OF TESTS DO YOU WISH TO ENTER
ENTER A NUMBER BETWEEN 1 AND 1
```

If there had been three tests already on file, the computer would have asked you to enter a number between 1 and 4. This would have allowed you not only to enter a new test, but also to re-enter any previous test. You might want to do this if you decide to give a test over for some reason.

If you have changed your mind about entering a test or if you hit the T key accidentally, don't worry. If you press RETURN instead of entering a test number, the program will return to the main menu. If you give a valid test number, you will be asked a series of questions. The first of these questions is:

```
WHAT DO YOU WISH TO CALL THIS TEST
(EXACTLY 3 NON-SPACE CHARACTERS PLEASE)
```

The three-character limit is imposed to provide the maximum space for reports, as we will see in a later chapter. You should use abbreviations that will help you remember what each set of grades is. Below are some examples of assignments and some appropriate abbreviations.

FIRST TEST	TS1
MIDTERM	MID
FINAL EXAM	FIN
LAB 2	L 2
LAB 12	L12

Actually, the non-space requirement only applies to the first and last characters as demonstrated by the LAB 2 assignment. The next question you will be asked is:

```
WHAT IS THE WEIGHT OF THIS TEST
```

Before you answer this question, let me explain how my program uses your answer. To begin with, the weights do not have to add up to 100 as you might suspect. Let's assume that you have entered three tests with the following weights.

TEST	WEIGHT
TS1	10
TS2	10
TS3	15

The Gradebook Program will perform the following calculations. It first adds the individual test weights to get a total of 35. The first two test scores for each student will be multiplied by 10/35. The third test will be multiplied by 15/35. The average is calculated by adding these adjusted scores.

This method for calculating averages has several advantages. First, even if all the possible scores have not been entered, the averages are still correct for present grades. This means that it is very easy to see how well a student is doing at any time. A second advantage is that the system is very flexible. You do not have to decide in advance exactly how many homework assignments, tests, etc., you are going to give. You can also change how much a test is worth by changing its weight. Chapter 6 will discuss how to make such changes.

For the sake of this example, enter a weight of 10 in response to the last question. The next question asked by the computer will be:

```
DO YOU WISH TO ENTER FROM THE CARD
READER (R) OR FROM THE KEYBOARD (K)
```

The use of the card reader will be discussed in Chapter 8, so press K. The program will respond with:

```
DO YOU WISH TO USE THE HIDDEN NUMERIC
PAD (Y/N)
```

If you choose N, then you can enter the grades using the normal keyboard layout. An answer of Y will cause the keyboard layout to be defined as shown below.

OLD KEY	NEW USE
J	1
K	2
L	3
U	4
I	5
O	6
7	7
8	8
9	9
SPACE	0
RTN	RTN
<	BACK UP

If you look at your keyboard, you can see that the above keys form a square similar to a numeric pad. Small labels can be added to your key caps to help you remember them if you wish. With a little practice, this arrangement can really speed up the manual entry of grades.

The hidden numeric pad enters numbers just as you would expect. The correction of errors is slightly different. If you make a mistake, you may start over by hitting any key other than those listed above. Such an action will cause a beep and erase the present entry.

No matter which keyboard you choose, the computer will print the name of the first student—in our example, the name DOE JANE. Enter a score of 80 for Jane. As each new name appears, enter the following grades: 87, 99, 63, and 78. You may enter the symbol < to back up and correct a previous grade just as when entering names. After entering the last grade, the system will automatically return to the menu.

If you feel unsure of any procedures, you may want to review some of the previous material. For those of you that are confident, though, please enter a second set of test scores. This second test should be called TS2 and should also have a weight of 10. The scores in alphabetical order are 55, 44, 88, 73, and 71. After entering the grades, save them to the disk. In the next chapter, I will show you how to examine these grades.

Chapter 5

EXAMINING GRADES

If you do not have the example file (MATH 101) of five students in memory, load it from the disk. The objective now is to see how we can examine the data that has been entered. From the menu press the A key. The program will respond by listing each of the students, his or her present average in the course, and the corresponding letter grade. The letter grades are assigned as follows:

A	90-100
B	80-89
C	70-79
D	60-69
F	0-59

You may use any break points you wish for your letter grades by altering lines 15060 through 15090 in the Gradebook Program. For example, if you would like the lowest grade for an A to be 94 instead of 90, change line 15060 to read:

```
15060 IF AVE(SN) >= 94 THEN PRINT "A": RETURN
```

Press any key to return to the menu.

Selecting the G option will list not only the averages, but also all of the grades for each student. If you have a large number of grades

entered for each student, the scores will wrap around the screen. This is not a problem, however, because this type of report is usually sent to a printer, not to the screen. If you have a printer you can use it for the following examples.

In order for the program to work as I wrote it you will need to have the interface card for the printer in slot 1 of your Apple. If your card is in some other slot you may either move it (turn power off) or change line 18030 in the Gradebook Program. When you are ready to use the printer, select option P from the menu.

Notice that each time P is pressed the words ON and OFF toggle in the print option. Also notice that the printer does not come on. Selecting the printer will only direct reports to the screen. Items such as the menu and prompts will not be printed.

With the printer option showing ON, press the G key. Because the printer has been selected the program will ask for some additional information. The first question will be:

```
INPUT YOUR HEADING BELOW
```

If you do not desire to put any special heading on your report, just hit RETURN. An example of what you might want to use might be your name and the date. You may use any characters except commas and semicolons. The computer will automatically include the present file name in the heading. An example might be:

```
BLANKENSHIP 10/14/84          MATH 101
```

The computer will then ask you:

```
HOW MANY SPACES BETWEEN LINES
```

If you want a single-spaced report, answer 0. A 1 will give you double space. If you want to give each student a copy of his or her grades, you might want to select 7 or 8 spaces between lines. This would provide ample room to cut the listing into pieces with a pair of scissors.

The next question will be:

```
HOW MANY NAMES DO YOU WANT ON EACH PAGE
```

If you hit RETURN, the names will be printed continuously. If you answer a number such as 35, a form feed will be issued to your printer every 35 names. These small features are not meant to provide a super deluxe formatting system. They will, however, provide an easy, workable approach to the type of reports needed for grades.

Sometimes, during a long listing of grades, you may wish to stop the scrolling. You may do so by pressing a control S. If you hit any other key during the listing, it will be aborted and the menu will appear when a second key is pressed.

Once you are through experimenting with the options discussed so far, return to the menu, toggle the printer off, and press the B key. The computer will respond by printing:

```
YOU CAN NOW LIST ALL STUDENTS BETWEEN
ANY LOW AND HIGH

ENTER THE NUMBER OF THE TEST YOU WISH
TO USE (OR 'A' TO USE THE AVERAGES)
```

This is self-explanatory. For our example press A. The program will then ask:

```
WHAT LOW AND HIGH DO YOU WISH TO USE
ENTER (LOW,HIGH)
```

If you want a list of all students whose present averages are in the 70's or 80's, enter 70, 90. The appropriate students and their grades will be listed on the screen. Even if you select a test, all of the grades will be listed. This can be very helpful. For example, if you list all the students that failed a test and see that some of them have high marks on other tests, you can bet that something is probably wrong.

Naturally, if you toggle the printer ON before asking for this report, the information will be sent to the printer instead of the screen.

Sometimes you might wish to examine only the grades of one particular student. You may do so by pressing F from the menu. The computer will respond by asking:

```
ENTER THE STUDENT'S NAME OR NUMBER
```

Not only may you use either a name or a number, but even partial names will work. You may, for example, find the grades for WHITE WENDY by entering just WHITE or, for that matter, just W. However, if you abbreviate too much you may not get the student you desire. Since there are two students that begin with SM (SMITH and SMART), entering SM will give you Smart's grades. This is true because the names are in alphabetical order and SMART was found first.

Now that you can examine the grades, the next chapter will show you how to fix any errors that you might discover.

Chapter 6

MAKING MODIFICATIONS

The ability to make modifications to any base of data is obviously essential. In this chapter, I will show you how easily you can modify elements in the Gradebook Program. Let's begin by modifying the weights associated with the assignments. Load the example class file that we used in previous chapters and select the W option from the menu. The computer will respond with:

```
HERE ARE THE PRESENT WEIGHTS
1.  (TS1) 10
2.  (TS2) 10

WHICH FIELD TO MODIFY
```

For our example, choose the second test by typing a 2 and hit RETURN. The computer will ask you to:

```
ENTER NEW TS2
```

Again, for the sake of example, enter a weight of 20. The new weights will automatically be printed on the screen. You could now enter another change, but since we are done, just press RETURN. If you wish, list the grades to see the effect this change has on the averages.

Now that you understand how weights may be changed, modifications for an individual student will be easy. Press M to select this option. The program will respond with:

WHAT IS THE NAME OR NUMBERS OF THE
STUDENT THAT YOU WISH TO MODIFY

Enter SMI and the screen will show:

1. SMITH JOHN
2. (TS1) 63
3. (TS2) 73

WHICH FIELD

Enter a 2 to select test 1 and then enter a 90 for the new grade. There are several special modifications that need to be mentioned at this time. If you change a grade to a −1, it will be treated as an excused test. Excused here means that the test in question will have absolutely no effect on the student's average.

Another special modification is to make the first letter of the name an asterisk. For example, change SMITH JOHN to *SMITH JOHN. Doing this effectively drops Mr. Smith from the class. He will no longer show up in the listings and his grades will not effect such things as the class average. You may retrieve the grades at any time, though, by using the M option to change the name back to normal. To do so, you must modify student number 4 because the name is no longer valid.

Try making some modifications and observe the effect they have on the other options we have discussed so far. Don't save your modified data, though, because we will use the original grades in the next chapter when we learn how to curve the grades.

Chapter 7

ANALYZING AND ADJUSTING GRADES

As any teacher knows, recording the grades is only the beginning. An analysis of the grades and their distribution is really a necessity. Since even the best of teachers can create a less than perfect test, the analysis can help to determine the cause of grades.

My Gradebook System can provide this analysis quickly and easily. To see how to accomplish this, run the program and LOAD the example class grades for MATH 101. From the main menu, choose the H option. The computer will print the following message.

```
YOU MAY NOW GET A STATISTICAL ANALYSIS FOR
ANY TEST OR FOR THE PRESENT AVERAGES
ENTER EITHER A TEST NUMBER OR AN 'A' FOR
AVERAGES
```

For example purposes, enter a 2 to get an analysis for test 2. The program will respond by printing the following information.

```
TEST #2  MATH 101
THE AVERAGE ON TEST #2 IS 66

90
80 * (1)
70 ** (2)
60
```

```
50 * (1)
40 * (1)
30
20
10
<9
THE LOWEST GRADE WAS 44
THE HIGHEST GRADE WAS 88
```

Remember, if you have selected the ON mode for the printer, you can get the above report as hard copy.

As you can see, you get specific information such as the high and low as well as the average grade. The distribution could have been shown using other methods, but I feel the histogram provides more information for most people. It is easy to see that no students were between 90 and 100. One made a grade in the 80's, two in the 70's, and so forth.

If you were to choose to curve these grades you have several options. To see these, press K from the menu. The screen will show.

```
WHICH TEST DO YOU WISH TO CURVE?
ENTER A NUMBER BETWEEN 1 AND 2
```

You have noticed no doubt, that you cannot curve the averages. The lack of such an option reflects my bias in this area. If a class is doing poorly throughout the term, it is de-motivational to the students to wait until the end to adjust the grades dramatically. A student should be able to determine his status at any time. Therefore, I feel that if curves are going to be applied, they should only be applied to individual tests.

Let's curve the test we previously analyzed by entering the number 2. The computer will respond with:

```
AUTOMATIC OR MANUAL (A/M)
```

Choose A for automatic. The screen will show:

```
THE GRADES WILL AUTOMATICALLY BE CURVED FOR TEST 2
THE PRESENT STATISTICS ARE
      THE AVERAGE IS 66
      THE LOW GRADE IS 44
      THE HIGH GRADE IS 88
WHAT DO YOU WANT THE NEW HIGH SCORE TO BE
```

For this example enter 93. The program will print:

```
YOU MAY NOW ENTER EITHER THE NEW LOW
SCORE OR THE NEW AVERAGE
WHICH DO YOU PREFER (L/A)
```

Press the L key and the computer will ask:

```
ENTER THE NEW LOW SCORE
```

Enter a score of 60. The computer will flash the word WORKING for a few seconds and then the following will appear.

```
THE CALCULATED VALUES FOR CURVING ARE
     A=.75
     B=27
PRESS N TO ABORT, ANY OTHER KEY TO
CONTINUE
```

You do not need to use these parameters. They will be used automatically if you press any key other than N.

When the menu returns, select the H option to see how the grades were effected by the curve. You should see the following histogram.

```
THE NEW ANALYSIS FOR TEST 2 IS

TEST #2  MATH 101
THE AVERAGE ON TEST #2 IS 77
90 * (1)
80 ** (2)
70
60 ** (2)
50
40
30
20
10
<9
THE LOWEST GRADE IS 60
THE HIGHEST GRADE IS 93
```

As you can see, the curve has been implemented correctly. Hit any key to return to the menu. You can see all the grades by using the G option.

You had two other choices when curving grades. In one of these you could have chosen the new high grade and a new average. Try this option at your leisure. The second choice was to select the manual mode. If you had chosen this option, the computer would have printed:

```
WE ARE READY TO CURVE TEST 2
PRESENTLY, THE STATISTICS ARE
      THE AVERAGE IS 66
      THE LOW GRADE IS 44
      THE HIGH GRADE IS 88
THIS IS THE CURVING FUNCTION
      NEW GRADE=(A*G)+B
G = THE GRADES TO BE CURVED
A = THE MULTIPLIER
B = THE ADDITIONAL POINTS
DO YOU STILL WISH TO CURVE (Y/N)
```

If you choose Y you will be asked to enter the values for A and B and the curve will be applied as in the automatic mode. This completes the curving options. In many cases, teachers prefer alternate methods of adjusting grades.

Two of these methods are dropping the lowest grade or grades, or replacing the lowest grade with another grade. For example, perhaps you have given 10 homework assignments during the term. You might want to drop the lowest three grades from each student's scores.

As a second example, assume that you have given three tests and a final in your course. You might feel that everyone has an off day occasionally and wish to replace the lowest test score with the final exam if it is higher. Both of these functions can be easily applied using the Gradebook Program. In keeping with my philosophy of keeping the students informed of their position in the class, I suggest that your policies concerning dropping and replacing be announced at the beginning of the term.

In order to have enough grades to demonstrate properly, we need to enter a third test. LOAD the original MATH 101 grades we have been using. Enter a final exam with a weight of 20 made up of the following grades: 92, 65, 93, 72, and 75. If you select the G option you should see the following scores.

NUM.	NAME	TS1	TS2	FIN	AVE
1.	DOE JANE	80	68	92	83
2.	JONES TOM	87	60	65	69
3.	SMART SAM	99	93	93	95
4.	SMITH JOHN	63	82	72	72
5.	WHITE WENDY	78	80	60	70

Save these grades to use later. After saving, replace the lowest of the first two grades with the final. Select the R option. The computer will respond with:

```
THIS MODULE WILL REPLACE THE LOWEST
GRADE OF SEVERAL TESTS WITH A TEST OF
YOUR CHOICE–IF IT IS HIGHER

INPUT THE TESTS TO BE CONSIDERED
(RETURN WHEN DONE)
```

Enter the numbers 1 (RTN) and 2 (RTN) and then return. The screen will show:

```
WHAT TEST NUMBER SHOULD BE USED AS THE
REPLACEMENT
```

Enter a 3 to indicate the final. When the menu returns, select the G option to see the grades again. They should look like this.

NUM.	NAME	TS1	TS2	FIN	AVE
1.	DOE JANE	80	92	92	89
2.	JONES TOM	87	65	65	71
3.	SMART SAM	99	93	93	95
4.	SMITH JOHN	72	82	72	75
5.	WHITE WENDY	78	80	60	70

Notice that the appropriate grades have been replaced with the final. Once you are finished with this example, LOAD the original grades. Press an O for ORDER and you will see:

```
THIS MODULE ALLOWS YOU TO SORT SEVERAL
GRADES SO THAT YOU CAN MODIFY WEIGHTS
TO DROP THE LOWEST ONES

ENTER THE TEST NUMBER OF EACH TEST TO BE
INCLUDED (RETURN WHEN DONE)
```

In our example, enter a 1 (RTN) and a 2 (RTN) and then just a RETURN. When the menu appears, list the grades using the G option. You should see:

NUM.	NAME	TS1	TS2	FIN	AVE
1.	DOE JANE	68	80	92	83
2.	JONES TOM	60	87	65	69
3.	SMART SAM	93	99	93	95
4.	SMITH JOHN	63	82	72	72
5.	WHITE WENDY	78	80	60	70

Notice that TS1 and TS2 are ordered. TS1 is always the lower of the two grades. If you wish to drop the lowest grade, you may now use the W option to change the weight of TS1 to zero.

In both replacing and ordering, the tests (or quizzes or homework assignments) being used should all have the same weight. Obviously, you would not want to drop the lowest of several grades if some are daily assignments and some are major tests.

You now have three options to alter the grades in a class. You can curve the grades, drop the lowest grades, or replace the lowest grade with another grade. You may not wish to use all of these, but almost everyone should be able to find a method that they find suitable.

If you perform a curve and then decide not to accept it, just LOAD the original file again from the disk. Some of you may wonder why I did not add an option to apply the curve in reverse to restore the grades. The reason is simple, although not very obvious. All the grades in my program are stored as integers to conserve memory space. The round-off errors when curving and uncurving can slightly alter the original grades. Several attempts to curve might cause major errors in the grades. My method, of course, causes no such problems and is generally very acceptable.

Let me remind you of one more item that can add some additional flexibility. If you use the M option in the menu, you can modify any grade. If you change any grade to a −1, it will be treated as an excused absence. It will have no effect, good or bad, on that student's average.

If you use the ORDER or REPLACE options on a set of grades that contains scores that are −1, you will get the following results. In order to keep excused grades from affecting the average, a score of −1 will be treated as if it was a 100. This means that if you are dropping the lowest grade for a student and he has an excused grade, the excused grade will not be the one dropped. If you don't like this idea, you have other options. You may simply not use grades lower than zero or you can delete lines 29624 and 29625 in the Gradebook Program. If these lines are deleted, the sort will treat grades of −1 as low grades.

A similar situation occurs when replacing grades. A −1 will never be replaced since it appears to this module as a 100. Again, if this is a problem, use a grade of 0 rather than a −1 or change lines 30230 and 30250 to read:

```
30230 IF G%(SN,TS(TL)) > G%(SN,TS(1)THEN TL=1
30250 IF G%(SN,TS(TL)) < G%(SN,TR)THEN G%(SN,TS(TL)) =
G%(SN,TR)
```

and delete line 30225.

At this point, you know how to use almost every option in the Gradebook System. The only capability not discussed so far is the use of the card reader. Since it is not essential for the operation of the system, the card reader was not discussed with the other options. If you plan to use it, just turn to the next chapter.

Chapter 8

USING THE CARD READER

Using the Gradebook Program without a card reader can save you hours of work. Using it with a card reader can make you feel like you just received a vacation. When I first started using it, I could not believe the difference. It was so efficient that I actually had enough time to make up tests the way they should be made up. And the time to do the job right is something sought after by every teacher.

In order to use the program as written, you need to acquire an OMR-2000 card reader for your computer and install it properly. The reader is available from:

Chatsworth Data Corporation
20710 Lassen Street
Chatsworth, California 91311
(213) 341-9200

The reader is reasonably inexpensive. It feeds the cards to be read automatically and can grade a class of 50 students in only a few minutes. It has been my experience that a school needs to have one reader for each 20 to 30 faculty members. It helps to have another computer or two available without a card reader to handle other gradebook functions.

Look first at Figure 8-1 to see what a card looks like. The student must fill in his or her student number at the top of the card. On this example card, Sam's number is 3. It should be left-justified as shown.

The number used must be the same one that appears by Sam's name when a gradebook report is generated.

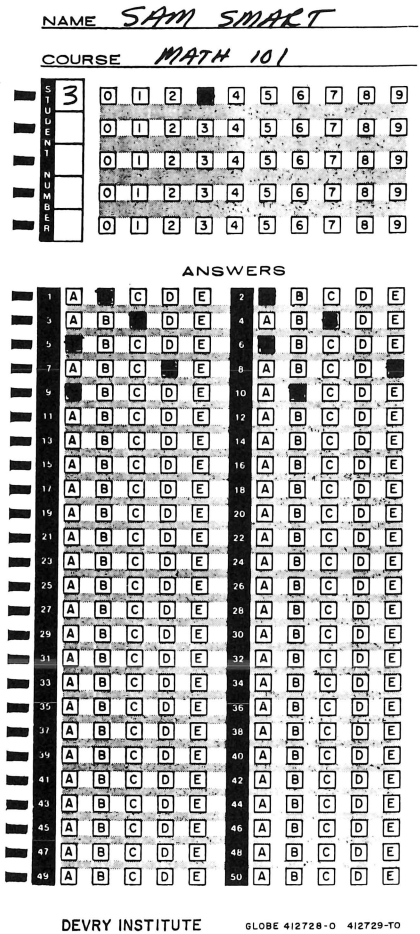


Figure 8.1

When I first enter the names into the computer I use the G option to get a list of the names and their associated numbers. I keep this list in my notebook so that forgetful students may check their number when they turn in their card. In order to help prevent errors, you should not change a student's number during the term. After all, students may have as many numbers as they have classes. To make sure that a number is not changed, my program will not allow you to sort the names after a test has been entered. If you feel you must perform such sorts, refer to Chapter 10.

The student number has two additional uses. A number of 88888 must appear on the last (top) card in the deck to tell the program when to stop reading. If the number is 99999 it means the card is the key to the test. A standard key is created by marking all the right answers. If you leave any answers blank, that question will be counted right for each student.

If you mark any two answers on the key for a question, it will be treated as a partial credit question. The best way to understand partial credit questions is with an example. Suppose you have a test with 22 multiple choice questions and two essay questions. Also assume that you wish the essay portion to count 50% of the test.

When making up the test, assign the essay questions the numbers 23 and 24 so that they will be at the end of the test and easy to locate. When you make up the key, fill in the correct answers for the multiple choice portion and any two answers for both questions 23 and 24. Naturally, you must grade the essays yourself, but my program will combine the scores for the two portions of the test. It will also enter the grades into the file.

Here's how it's done. The students should mark the answers to the multiple choice questions on the cards, but leave questions 23 and 24 blank. Have the students fold their papers and place the card inside. This keeps them in order and makes things easy for you. When you read the essays, determine how much credit you think the student deserves. Mark questions 23 and 24 yourself, using the following chart to give the credit you wish.

A	100%
B	75%
C	50%
D	25%
E	0%

Some teachers I have talked with prefer to have ratings of 100, 90, 80, 70, and 0. You may set any ratings you prefer by adjusting lines 33210 through 33230 in my program. Simply replace each occurrence of the numbers .75, .5, and .25 with the numbers you desire. Note that the numbers appear twice in each line.

Partial credit questions can be any weight relative to the regular questions. Each partial credit question must, however, have the same weight. In the previous example, each partial credit question was worth eleven times as much as each multiple choice question. In some cases, it may be wise to break a large essay question into several smaller questions. This should make it easier to distribute points as you see fit.

If your test has several short answers or fill-in-the-blank questions, you may wish to have the student write the answers on the back of the card. This means that you can let the students keep the test and only turn in the card. I rather enjoy not having to carry pounds of paper around with me.

Some people complain that small inexpensive card readers are not accurate. I have found this to be untrue. Errors are most frequently due to poorly filled-out cards. I can offer a couple of suggestions, however, that can induce students to fill out the cards carefully. Whenever I have a new class, I like to have them fill out cards for a "practice" test. I give them the answers on the key and let them try light and dark marks, partial erasures, or anything they wish. When they get the results back, they know what they can and cannot expect from the card reader.

In addition, I have a standing rule that questions that are wrong due to obvious carelessness on the part of the student will not be corrected. I defend this position because I think it is important for students to learn to follow directions and assume the responsibilities of their actions. It is surprising how well students can fill out a card if they know they have to.

Lastly, I attempt to make the card reader totally infallible by having it stop the grading process and ask you what a particular answer or student number was on a card that is not readable. Naturally, such a process may not always work. Nothing is ever perfect, but this method does ensure that student numbers will be in range and that most poor erasures will be caught. If you are asked by the computer to give the correct answer to a question, look at the card that has just been ejected from the reader and enter the answer marked. If the student failed to mark the card or made multiple marks, enter a dash (-).

Now that you know how to create the cards, prepare a sample one for each of the five students in our mythical MATH 101. Use the student numbers 1 through 5 as seen in the reports. Place your key on the bottom of the deck and the card indicating the end on the top. Place the cards in the hopper and turn the reader on as indicated in the Chatsworth manual. Make sure you have entered the MATH file from the disk and then choose the T option.

Since we have already entered three tests, ask for test 4. Assign it a name and weight of your choice. When asked if you wish to enter from the keyboard or the reader, press R. The computer will respond with:

```
MAKE SURE THE CARD READER IS TURNED ON
AND THAT THE CARDS (IN THE PROPER ORDER)
ARE IN THE HOPPER
```

```
HOW MANY QUESTIONS ARE ON THIS TEST
(50 MAX)
```

Enter the number of questions on your sample test. The computer will then ask.

```
IF THERE ARE ANY PARTIAL CREDIT PROBLEMS
ON THIS TEST WHAT IS THEIR RELATIVE
WORTH
```

If you have none or if the relative worth is 1, just press RETURN. Otherwise, enter the correct weight. If you have the printer toggled ON, you will be asked to prepare the printer for the individual grade reports. The reason for this is that the output is formatted for labels if you wish to use them. Hit RETURN when you are ready for the computer to begin reading the cards.

As each card is read and graded, the questions missed by each student are printed beneath his or her name. Partial credit questions are handled in a manner that indicates the amount of credit given. For example, 12.75 would indicate that the student received 75% credit on question 12. Figure 8.2 is an output of the results of some sample cards.

When the cards have all been graded, the computer will tell you:

```
PREPARE PRINTER FOR ANALYSIS
```

This gives you a chance to change from labels, if you were using them, to regular paper. A sample test analysis can be seen in Figure 8.3 As you can see, you can determine how many people chose each answer, what the correct answer was, and the percentage of people getting the question right.

If more than 80% get a question right or wrong, it is flagged as EASY or HARD. The analysis ends with a statement of the percentage of the multiple choice questions that were easy and hard. This information can be of great help when you make up new tests.

Any student not turning in a card will receive a grade of −1. Use the modify function to alter these grades as you see fit.

```
1.DOE JANE
1 6
GRADE = 80


5.WHITE WENDY
1 4 5 8 9.5 10
GRADE = 45


4.SMITH JOHN
1
GRADE = 90


3.SMART SAM
2 4 10.75
GRADE = 78


2.JONES TOM
1 9.5
GRADE = 85
```

Figure 8.2

```
THE LOWEST GRADE IS 45
THE HIGHEST GRADE IS 90

ANALYSIS FOR TEST 4 - MATH 101


QUES.   A   B   C   D   E   CORRECT   %RIGHT

 1      1   1   2   0   1      B        20   HARD
 2      1   0   4   0   0      C        80   EASY
 3      0   0   5   0   0      C       100   EASY
 4      0   1   1   3   0      D        60
 5      4   1   0   0   0      A        80   EASY
 6      4   1   0   0   0      A        80   EASY
 7      0   0   0   5   0      D       100   EASY
 8      0   0   0   1   4      E        80   EASY
 9      3   0   2   0   0      *    NOT MULTIPLE CHOICE
10      3   1   0   0   1      *    NOT MULTIPLE CHOICE


60% OF THE QUESTIONS WERE EASY
10% OF THE QUESTIONS WERE HARD
```

Figure 8.3

Chapter 9

MODULE DESCRIPTIONS

No matter how good a program is, it can never contain all the features wanted or needed by the people that end up using it. For that reason, the diskette included with this book has no copy protection on it. You can easily make backups, as well as list and modify the code. I feel it is my responsibility to ensure that you have these capabilities. If you agree with these principles, you need to support programmers such as myself by encouraging people that wish to make copies of this program to buy the documentation. I have spent considerable effort to make the documentation worth buying.

One of the important parts of documentation that, oddly enough, is usually left out is the information needed to make program modifications. For your convenience, Appendix A contains a complete listing of the Gradebook Program. Look quickly at the program and you will see that it is divided into small sections with descriptive headers.

Each of these sections is called a module. In this chapter I will provide a short description of each of these modules. I will try to give you enough information so that you can customize your copy to meet your specific needs. In some cases the title of a module is all that is needed for an explanation but, to be consistent, I will cover every module in the order in which they occur in the program. Note that this order has no bearing on the order in which they are executed.

Before I discuss the modules, let me describe briefly each of the variables used in the Gradebook Program. Your knowledge of these

variables will help your understanding of how the modules work. The variables used, in alphabetical order, are:

A	General purpose.
A$	General purpose.
AG	Flag to indicate if grades are averaged; =0 if not averaged, =1 if they are.
ALL	Flag to indicate if only averages or all grades are to be printed. 0 means only averages, 1 indicates all grades.
AVE	Temporary variable used when calculating student averages.
AVE(	Holds average of each student.
B	General purpose.
B$	General purpose.
C$	Holds the single-letter commands in the menu.
CR	General purpose.
CURVE(	Function variable holding the curving equation.
D$	Used for a control D for disk operations.
FL$	Present file name.
G	Old grade in curving function.
G%(	Two-dimensional array holding the tests (TN) for each student (SN). Typically G%(SN,TN).
HD$	Heading used on reports.
HG	High grade when analyzing tests.
I	General purpose.
J	General purpose.
K	General purpose.
K$	Holds the key for the test being graded.
LG	Low grade when analyzing tests. LP = NUMBER OF STUDENTS PER PAGE TO PRINT
N$	Present name being manipulated.
N0	Number of students between 0 and 9 during analysis.
N10	Number of students between 10 and 19.
N20	Number of students between 20 and 29.
N30	Number of students between 30 and 39.
N40	Number of students between 40 and 49.
N50	Number of students between 50 and 59.
NA	Number of students between 90 and 100.

NA$(	Sometimes NAME$(,. Holds names of students.
NB	Number of students between 80 and 89.
NC	Number of students between 70 and 79.
ND	Number of students between 60 and 69.
NE	Number of easy questions on a test.
NH	Number of hard questions.
NL	Number of lines printed on a label so far.
NM	Number of questions missed on a test.
NS	Number of students in the present class.
NT	Number of tests presently in memory.
PC	Value of partial credit questions.
PER(	Array containing weights of each test.
PF$	Printer flag. Either ON or OFF.
R$	Holds data from last card read.
RQ	Number of regular (not partial credit) questions.
SN	Number of the student being manipulated.
SP	General purpose.
SP$	Number of spaces between lines of a listing.
T	General purpose.
T$	General purpose.
TA	Average score on a test.
TH$(	Array holding names of the tests.
TL	Index to the lowest test grade.
TN	Number of test being manipulated.
TP	Total points possible for a student so far.
TR	Index to replacement test.
TS	Index for tests being sorted.
TS(	Array holding indexes of the tests to be sorted.
TV	Total of the grades that are void or excused.
WW	Number of questions on each line of a label.
X$	General purpose.

Now let's look at the descriptions for the modules.

MAIN PROGRAM

This module uses other modules to initialize the system and to print the menu. It inputs your choice and directs the program to one of the main subroutine modules that can carry out the desired action.

PRINT MENU

In addition to printing the menu, two other actions are performed. One of these is the resetting of the ONERR flag. In most cases each module of the Gradebook System will not allow any actions that could potentially cause an error. The nature of the LOAD and SAVE modules requires the use of the ONERR statement. Since no other module needs or uses it, the flag is reset each time the menu is printed.

The second action is to turn off the printer. When the printer flag is on, each module that uses the printer will turn it on. Rather than have each module worry about turning the printer off, it is handled here.

TOGGLE PRINTER

This module toggles the printer flag (PF$). Other modules check to see if this variable contains the characters ON or OFF to decide whether the printer should be enabled.

CATALOG

This module will perform a catalog or directory command. It allows you to see either Drive 1 or Drive 2.

EVERYBODY'S AVERAGE, EVERYBODY'S GRADES

These two modules are almost identical. They both input a heading if the printer is ON. Each of them calls 8000 to actually list the names. The difference is that if only averages are desired, the variable ALL is set to 0. If all the grades are wanted ALL is set to 1.

LIST THOSE BETWEEN

As you would expect, this module inputs two numbers, R1 and R2, and lists all of the students and their grades whose average or selected test score is between R1 and R2.

LIST EVERYONE

A loop is set up to create the student numbers (SN) from 1 to the number of students (NS). For each SN, the module at 9000 is called to list

for that student. Line 8026 is used to see if a key is pressed. If a key is pressed, the listing is aborted. The variable AG is set to 1 to indicate that the grades have been averaged.

LIST FOR STUDENT SN

Three modules are called to print the name, grades, and average for each student. The flag AG is also checked to see if the grades have previously been averaged. If AG=1, the grades have been averaged. If AG=0, the module at 11000 will average the grades for the student being listed.

The variable SP$ is used to determine how many lines to skip between names. Names that begin with * are not listed.

LIST BY NAME OR NUMBER

If a student number was entered, this module will check to see if it is within the valid limits. If a name was input, the number is looked up using the subroutine at 16000. The titles of the tests are printed using 36000.

AVERAGE GRADES FOR STUDENT SN

This module determines the total possible points (TP). Each grade is multiplied by its weight PER(TN) and divided by the total points. TN indicates the number of the test being dealt with. The weighted grades are added together to form the average. Each average is rounded off appropriately and stored in AVE(SN) where SN is the student number.

PRINT THE NAME OF STUDENT SN

This subroutine expects SN to be set before entry. It prints the number and the name truncated to 15 characters. You may want to reduce the allowable characters to make room for more columns of tests on printouts.

PRINT GRADES AS 3 CHARACTERS

As the title indicates, this module makes each grade exactly three characters for formatting purposes. Each grade is printed on the same line with no carriage return issued.

PRINT AVERAGE

Like the last module, the average is converted to exactly three characters for formatting purposes. It is printed in inverse to make it stand out. The letter grades are then printed and a carriage return is sent to set up the next line. You may alter the scores required for the letter grades by placing the lowest acceptable score for each grade in lines 15060 to 15090.

SEARCH NAMES TO FIND SN

When entered, this module expects NM$ to contain the name being searched for. SN is set to the number of the name if found, or to 0 if it is not found.

INITIALIZATION

The arrays are dimensioned to allow for 125 (124 if using card reader) students and 25 grades per student. Each test (using the card readers) can have 50 questions. The printer flag is set to OFF. The remainder of the routine sets up the required ERROR detecting routines.

TURN PRINTER ON

This module does exactly that. This line should be changed to reflect the proper way to turn your printer on.

ENTER NEW NAMES

This module reacts differently, depending on the number of students (NS) presently entered. As the names are entered, they are checked for a null name to quit and for a < to back up to correct a previous name. Names can be sorted as long as no grades have been entered.

ENTER NEW GRADES

A check is made to ensure that names are entered before grades. You can enter any of the previous tests or the next one. You must enter a name for the test and its weight. If you desire to use the card reader,

another module is called to perform the task. If you enter the grades from the keyboard, you may back up by using that < symbol. Students that have been dropped by making the first character of their name an * are not available for entering grades.

MODIFY BY NAME OR NUMBER

If the entered item is a name, it is converted to the proper student number by calling 16000. Once a number is found, or if the original input was a number, MODIFY BY NUMBER is executed.

MODIFY BY NUMBER

Upon entry, the variable A contains the number of the student to be modified. The appropriate information is presented for modification until a null input is received. If you select a field to modify and then hit RETURN, the modification is aborted leaving the field unchanged.

MODIFY WEIGHTS

This module is identical in operation to MODIFY BY NUMBER, except that the weights of each test may be changed.

ENTER FROM DISK

DISK loads are checked for errors using the ONERR statement. The file asked for is OPENed and data is read in the following order. The number of students (NS) is the first item in the file. It is used to set the length of a loop to read the name of each of the students. The number of tests (NT) is then read and used to read the weight and heading as well as the scores for each test. The variable AG is set to 0 to indicate that the grades are not averaged.

SAVE TO DISK

The ONERR system is used to prevent failure if the disk operation fails. Examples of possible failures are locked files and full disks. The grades are saved in the same order as they are read (see ENTER FROM DISK).

STATISTICAL ANALYSIS

This section checks to see if you wish to analyze a TEST or the AVERAGES and transfers control to the appropriate module.

ANALYZE A TEST

When entered, TN contains the number of the test to be analyzed. A loop is set up to examine the test of each student. Counters keep track of the number of scores in each of 10 equal areas, such as between 90 and 100. In order to calculate the average grade, variable TV (tests that are invalid) keeps track of how many tests are excused or belong to a student that has been dropped. Each grade is checked against a high and low grade to establish the same. A check is made so that grades of —1 do not count as low grades.

PRINT I*'s

This section is used when printing histograms. It prints the numbers of asterisks indicated by the variable I. It also prints the number of asterisks in parentheses at the end of the line.

ANALYZE AVERAGES

This module performs the same operation as the ANALYZE TEST module except that it works on the average grades.

CURVE GRADES

This section determines if you wish to curve manually or automatically. If automatic is your choice, another module is executed. Otherwise, the linear function A*G+B is applied to each grade. The manual reference here means that you must supply the values for the variables A and B.

SORT FOR DROP

This module allows you to choose several sets of tests to be sorted. It treats grades of —1 as if they were the highest grade instead of the lowest.

REPLACE LOWEST GRADE

You may select several sets of tests to work with as well as a replacement test. The routine then finds the smallest test of the first group and replaces it, if the replacement is larger. Grades of —1 will never be replaced.

AUTOMATIC CURVING

This module allows you to enter a new high score and either a new average or a new low score. This information is used to calculate the appropriate A and B used in the formula for manual curving. The statistical module is called (using a specific flag to prevent printing to the screen) to get the data needed to perform these calculations. Once A and B are established, the program uses the last portion of the manual routine to apply the curve.

REPORT FORMAT

This section is called only if the printer flag is ON. It allows you to set the parameters to be used when listings are generated.

INPUT FROM CARD READER

This section not only inputs the data on the card, but also handles the grading of the test. Lines 33060 to 33101 perform the actual read of the card. If you are using a different card reader, replace these lines with a program to insert the card data into the string variable R$. The first five characters must make up the student number. The next 50 characters are the test answers.

The answers must be A, B, C, D, E, —, or *. The dash means that no answer was marked on the card while an asterisk means more than one choice was indicated. Whenever a student's card indicates an invalid student number or a double answer, you are asked to input the correct data.

Blank answers on the key mean that everyone gets the right answer. Double marks on the key indicate a partial credit question. Partial credit questions are handled by their own module.

PARTIAL CREDIT QUESTIONS

Partial credit questions have a weight specified by the variable PC. Answers on the student card have the following meaning:

A	100% correct
B	75% correct
C	50% correct
D	25% correct
E	Entirely wrong

You may alter these weights by changing the appropriate lines.

PRINT TEST RESULTS

This section reports the findings of the question analysis including the percentage of the test made up of hard and easy questions.

FLASH WORKING

Several routines throughout the program use this module to indicate when they are busy performing a task that may take some time.

PRINT TEST TITLES

This module prints a title line for reports, including the titles for each test.

ANSWER ANALYSIS

This routine prints the correct answer to each question and an indication of whether it is a hard or easy question. The program assumes that any question missed by 80% of the class is hard and any question answered correctly by 80% to be easy. You may modify lines 30780 and 30785 to reflect your bias if you wish. The variables NE and NH keep track of the number of easy and hard questions that occur.

Although most people will use the Gradebook System exactly as I wrote it, it is easy to utilize or modify the basic routines in the

program to satisfy any special needs you have. Fred Kerr, one of my fellow teachers at DeVry, has demonstrated this modification capability to an extreme. Using only the information in this chapter, he converted the Gradebook System into a weekly attendance system.

I have not thoroughly tested his program, but with his permission, I have provided a copy of it on the Gradebook Diskette. Since it is offered as a modification example, no documentation is provided. It is menu-driven, though, and operates much like the original Gradebook. It is a good demonstration of how easy it is to modify structured code.

Chapter 10

MODIFYING DATA FILES

In the last chapter, I tried to give you enough information about the Gradebook Program to allow you to customize it to meet your specific needs. Because of the amount of memory required by the program and its data, there is not a great deal of room left to add things. This chapter will show you how to get around the space limitation.

Since the Gradebook Program handles all of the normal activities that you might need, any major features that are desired will usually not be used on a day-to-day basis. Since such features will typically need to manipulate the data files, we can have a separate program read the same files, alter them as necessary, and then store the files back on the disk.

Before you start assuming that writing such a program is beyond your capabilities, consider this. The portions of the program needed to read and write the data files are already written. All you have to do to get them is to load the Gradebook Program and delete all the lines except the modules that you need.

To show you how easy it is to build extensions to the original program in this way, I have written a short example. The purpose of this program is to sort the class file using any field as the key. This allows you to alphabetize the students after grades have been entered. It also allows you to order the students by the grade that they made on any test or by their averages. In order to calculate the averages, I again borrowed a routine from the original Gradebook Program.

When you use this program, you should usually save the new file using a new name. Notice that this program cannot create any reports, it only modifies files. Once you have saved a file, you can RUN the original Gradebook Program and use it to list the students and their grades in the newly created order.

A listing of this program can be found in Appendix B. It is also found on the diskette under the name FLEXIBLE SORT. If you can program in BASIC, and if you study Chapter 9 and this example, you should have no trouble adding any features you desire to my Gradebook System.

Chapter 11

A FINAL WORD

In the previous chapters I have tried to show you how to get the most from my Gradebook Program. Each and every feature has been explained in detail so that even an inexperienced computer user should have no problem operating the system.

The program has been used by many teachers. Their inputs and suggestions have led to many modifications. I'm very proud of the finished product and feel it is the easiest to use and most versatile gradebook program I have seen.

I'm also aware that you can't please everyone, so I have tried to present enough information to allow anyone with even a moderate amount of programming experience to be able to perform custom modifications. There are very few programs available with this type of documentation.

Most gradebook programs sell for many times the price of *The Gradebook System.* Programmers typically raise the price of programs to make up for the piracy that is expected. I don't believe such actions are necessary. My intent from the beginning was to provide more to the purchaser than just a program.

I want to provide a modifiable disk, enough information to make modification possible, and a tutorial to get users up to speed quickly. And I wanted to do it at a price low enough that bootleg copies would be unnecessary. I think such a product can survive the perils of piracy. If you appreciate my philosophy then I ask you again to encourage other teachers to purchase the program.

I would like to close by offering to you my hope that *The Gradebook System* will help you become a better teacher by giving you more information about the tests you give now and more time to create new ones in the future.

APPENDICES

Appendix A

GRADEBOOK

```
10   REM !------------------------!
20   REM !  INSTRUCTORS GRADEBOOK  !
30   REM !         WRITTEN BY     !
40   REM !      JOHN BLANKENSHIP  !
50   REM !       DEVRY  ATLANTA   !
53   REM !   COPYRIGHT JULY 1983  !
60   REM !------------------------!
70   REM
80   REM !------------------------!
90   REM !       MAIN PROGRAM     !
95   REM !------------------------!
100   GOSUB 16500: REM  INITIALIZE
110   GOSUB 1000: REM  PRINT THE MENUE
120   VTAB 24: HTAB 1
121   PRINT "SELECT LETTER OF YOUR CHOICE    ";
122   HTAB 30
125   GET A$
130  C$ = "NTMWAGFBSLCKROHPE"
140   FOR A = 1 TO  LEN (C$)
150   IF A$ =  MID$ (C$,A,1) THEN 200
160   NEXT A
170   PRINT  CHR$ (7);: GOTO 120
200   ON A GOSUB 20000,21000,22000,24000,3000,4000,100
      00,5000,26000,25000,2500,28000,30000,29000,27000
      ,2000,17000
210   GOTO 110
1000   REM !-----------------!
1002   REM !  PRINT MENUE     !
1004   REM !-----------------!
1005   HOME
1006   POKE 216,0: REM RESET ONERR TO NORMAL
1008   PRINT : PRINT  CHR$ (4)"PR#0"
1010   INVERSE : HTAB 5: PRINT "THE COMPLETE GRADEBOOK
       SYSTEM": NORMAL
```

```
1020  PRINT "     BY PROFESSOR JOHN BLANKENSHIP"
1030  PRINT "     DEVRY INSTITUTE OF TECHNOLOGY"
1040  PRINT "          COPYRIGHT  JULY 1983"
1050  PRINT
1060  PRINT "N-NAMES (ENTER)"
1070  PRINT "T-TESTS QUIZES HOMEWORKS (ENTER)"
1080  PRINT "M-MODIFY DATA"
1090  PRINT "W-WEIGHTS (MODIFY)"
1100  PRINT "A-AVERAGES (LIST)"
1110  PRINT "G-GRADES (LIST)"
1120  PRINT "F-FIND ONE STUDENTS GRADES"
1130  PRINT "B-BETWEEN TWO GRADES (LIST)"
1140  PRINT "S-SAVE GRADES TO DISK"
1150  PRINT "L-LOAD GRADES FROM DISK"
1160  PRINT "C-CATALOG"
1170  PRINT "K-CURVE GRADES"
1180  PRINT "R-REPLACE LOWEST GRADE"
1190  PRINT "O-ORDER GRADES FOR DROPPING"
1200  PRINT "H-HISTOGRAM (STATISTICAL ANALYSIS)"
1210  PRINT "P-PRINTER (ON/OFF)   ";
1220  INVERSE : PRINT PF$: NORMAL
1230  PRINT "E-END PROGRAM"
1240  PRINT
1250  RETURN
2000  REM !----------------------!
2010  REM !     TOGGLE PRINTER    !
2020  REM !----------------------!
2030  IF PF$ = "ON" THEN PF$ = "OFF": RETURN
2040 PF$ = "ON": RETURN
2500  REM !------------------!
2502  REM !     CATALOG      !
2504  REM !------------------!
2505  ONERR  GOTO 2630
2506  HOME : PRINT "DRIVE 1 OR 2 ";: GET A$
2507  PRINT : HOME
2508 A =  VAL (A$)
2510  IF A = 1 OR A = 2 THEN  PRINT  CHR$ (4)"CATALOG
      D"A
2610  IF PF$ = "OFF" THEN  GET A$
2620  RETURN
2630  CALL 768: GOTO 2506
3000  REM !------------------!
3002  REM ! EVERYBODY'S AVE !
3004  REM !------------------!
3005  HOME :ALL = 0
3006  GOSUB 32000: PRINT : PRINT : GOSUB 36000
3020  GOSUB 8000: REM  LIST EVERYBODY
3025  IF PF$ = "OFF" THEN  GET A$
3030  RETURN
4000  REM !--------------------!
4002  REM ! EVERYBODY'S GRADES !
4004  REM !--------------------!
4005  HOME :ALL = 1
4006  GOSUB 32000: PRINT : PRINT : GOSUB 36000
4020  GOSUB 8000: REM  LIST EVERYBODY
4025  IF PF$ = "OFF" THEN  GET A$
4030  RETURN
5000  REM !--------------------!
5002  REM ! LIST THOSE BETWEEN !
5004  REM !--------------------!
5005  HOME :ALL = 1
```

```
5010  PRINT "YOU CAN NOW LIST ALL STUDENTS BETWEEN
    ANY LOW AND HIGH": PRINT
5011  INPUT "ENTER EITHER A TEST # OR AN 'A' IF YOU
    WANT TO USE THE AVERAGE ";A$
5012  PRINT : IF  NOT (A$ = "A" OR ( VAL (A$) > 0 AND
     VAL (A$) <  = NT)) THEN  RETURN
5015  INPUT "WHAT LOW AND HIGH DO YOU WISH (LOW,HIGH)
    ";R1,R2
5021  HOME : IF PF$ = "ON" THEN  GOSUB 18000
5022  PRINT "THE STUDENTS BETWEEN "R1" AND "R2
5023  IF A$ = "A" THEN  PRINT "FOR THEIR PRESENT AVER
    AGE ARE:": GOTO 5025
5024  PRINT "ON TEST # " VAL (A$)" ARE:"
5025  PRINT : GOSUB 36000
5026  IF NS = 0 OR NT = 0 THEN  RETURN
5027  FOR SN = 1 TO NS
5030  IF AG = 0 THEN  GOSUB 11000: REM  AVERAGE GRAD
    ES FOR STUDENT SN
5031 A = AVE(SN)
5032  IF A$ <  > "A" THEN A = G%(SN, VAL (A$))
5040  IF A >  = R1 AND A < R2 THEN  GOSUB 9000
5050  NEXT SN
5052 AG = 1
5055  IF PF$ = "OFF" THEN  GET A$
5060  RETURN
8000  REM !---------------!
8002  REM ! LIST EVERYONE !
8004  REM !---------------!
8005  IF NS = 0 THEN  PRINT "NO STUDENTS ON FILE": GET
    A$: RETURN
8006 NL = 0
8010  FOR SN = 1 TO NS
8020  GOSUB 9000: REM  LIST FOR STUDENT SN
8025 NL = NL + 1: IF NL = LP THEN NL = 0: PRINT  CHR$
    (12): REM FORM FEED
8026  IF  PEEK (49152) > 127 THEN  POKE 49168,1: RETURN
    : REM  CHECK FOR KEY PRESSED
8030  NEXT SN
8035 AG = 1
8040  RETURN
9000  REM !----------------------!
9002  REM ! LIST FOR STUDENT SN !
9004  REM !----------------------!
9005  IF  LEFT$ (NAME$(SN),1) = "*" THEN  RETURN
9006  GOSUB 13000: REM  PRINT NAME OF SN
9007  IF NT = 0 THEN  PRINT : GOTO 9040
9010  IF AG = 0 THEN  GOSUB 11000: REM  AVERAGE FOR
    STUDENT SN
9020  IF ALL = 1 THEN  GOSUB 14000: REM  PRINT GRADES
     FOR SN
9030  GOSUB 15000: REM  PRINT AVERAGE FOR SN
9040  IF  VAL (SP$) = 0 THEN  RETURN
9050  FOR SP = 1 TO  VAL (SP$): PRINT : NEXT SP
9060  RETURN
10000  REM !--------------!
10002  REM ! LIST BY NAME !
10003  REM !  OR  NUMBER  !
10004  REM !--------------!
10010  HOME
10020  INPUT "ENTER THE STUDENT'S NAME OR NUMBER
    ";NM$
```

```
10021  IF NM$ = "" THEN  RETURN
10022 ALL = 1
10023  GOSUB 36000
10025 SN =  VAL (NM$): IF SN > 0 AND SN <  = NS THEN
     10040
10030  GOSUB 16000: REM  FIND SN FOR NM$
10040  IF SN <  > 0 THEN  GOSUB 9000: REM  PRINT GRAD
     ES FOR SN
10045  GET X$
1000  RETURN
11000  REM !-------------------------!
11002  REM ! AVE GRADES FOR STUD SN !
11004  REM !-------------------------!
11005 TP = 0: FOR TN = 1 TO NT
11006  IF G%(SN,TN) < 0 THEN 11008
11007 TP = TP + PER(TN)
11008  NEXT TN
11010 AVE = 0
11020  FOR TN = 1 TO NT
11025  IF G%(SN,TN) < 0 THEN 11040
11030 AVE = AVE + (PER(TN) / TP) * G%(SN,TN)
11040  NEXT TN
11050 AVE(SN) =  INT (AVE + .5)
11060  RETURN
13000  REM !-------------------------!
13002  REM ! PRINT NAME OF STUD. SN !
13003  REM !   AS 15 CHARACTERS      !
13005  REM !-------------------------!
13006  IF SN < 10 THEN  PRINT " ";
13007  IF SN < 100 THEN  PRINT " ";
13008  PRINT SN". ";
13010 N$ = NAME$(SN)
13020  IF  LEN (N$) > 15 THEN N$ =  LEFT$ (N$,15)
13030  IF  LEN (N$) <  > 15 THEN N$ = N$ + " ": GOTO
     13030
13040  PRINT N$;" ";
13050  RETURN
14000  REM !-------------------------!
14002  REM ! PRINT GRADES AS 3 CHAR !
14004  REM !-------------------------!
14010  FOR TN = 1 TO NT
14020 T$ =  STR$ (G%(SN,TN))
14030  IF  LEN (T$) <  > 3 THEN T$ = " " + T$: GOTO 1
     4030
14040  PRINT T$ + " ";
14050  NEXT TN
14060  RETURN
15000  REM !-----------------------!
15002  REM ! PRINT AVE IN INVERSE !
15003  REM !  AND LETTER GRADE     !
15004  REM !-----------------------!
15010 T$ =  STR$ (AVE(SN))
15020  IF  LEN (T$) <  > 3 THEN T$ = " " + T$: GOTO 1
     5020
15030  INVERSE
15040  PRINT T$;
15050  NORMAL : PRINT " ";
15060  IF AVE(SN) >  = 90 THEN  PRINT "A": RETURN
15070  IF AVE(SN) >  = 80 THEN  PRINT "B": RETURN
15080  IF AVE(SN) >  = 70 THEN  PRINT "C": RETURN
15090  IF AVE(SN) >  = 60 THEN  PRINT "D": RETURN
15100  PRINT "F": RETURN
```

```
16000  REM !-------------------------!
16002  REM ! SEARCH NAMES TO FIND SN !
16004  REM !-------------------------!
16010  FOR SN = 1 TO NS
16020  IF NM$ =  LEFT$ (NAME$(SN), LEN (NM$)) THEN 16
     060
16030  NEXT SN
16040 SN = 0
16050  PRINT "THAT NAME WAS NOT FOUND"
16055  FOR I = 1 TO 1500: NEXT I
16060  RETURN
16500  REM !----------------!
16505  REM ! INITIALIZATION !
16510  REM !----------------!
16520  DIM G%(125,25),NAME$(125),PER(25),AVE(125),TS(
     25),K%(50,5),TH$(25)
16530 PF$ = "OFF"
16535  TEXT
16539  REM  ONERR ROUTINE
16540  FOR I = 1 TO 10
16550  READ A
16560  POKE 767 + I,A
16570  NEXT I
16580  RETURN
16600  DATA  104,168,104,166,223,154,72,152,72,96
17000  REM !--------------------!
17005  REM !  END OF PROGRAM    !
17010  REM !--------------------!
17020  HOME
17030  INPUT "ARE YOU SURE YOU WISH TO QUIT ?
      (YES OR NO) ";A$
17040  IF A$ = "YES" THEN  END
17050  RETURN
18000  REM !----------------------!
18010  REM !   TURN PRINTER ON    !
18020  REM !----------------------!
18030  PRINT : PRINT  CHR$ (4)"PR#1"
18040  RETURN
20000  REM !----------------------!
20010  REM !   ENTER NEW NAMES    !
20020  REM !----------------------!
20030  HOME
20031  IF NS > 0 THEN  PRINT "THERE ARE NOW "NS" STUD
     ENTS": PRINT "IN THE FILE": GOTO 20040
20032  PRINT "YOU MAY NOW ENTER NEW NAMES
      (MAX. OF 124 STUDENTS)":NT = 0:NS = 0: PRINT : GOTO
     20220
20040  PRINT "DO YOU WISH TO ENTER A NEW CLASS OR ADD
      TO THE PRESENT ONE (E/A) ";: GET A$
20050  IF A$ = "E" THEN 20100
20060  IF A$ = "A" THEN 20200
20070  RETURN
20100 NS = 0:NT = 0
20200  HOME
20210  PRINT "THERE ARE NOW "NS" NAMES"
20220  PRINT "ENTER EACH NAME - JUST RTN WHEN DONE"
20230  PRINT : PRINT
20240  PRINT NS + 1". ";
20250  INPUT NAME$(NS + 1)
20251  IF NAME$(NS + 1) <  > "<" THEN 20260
20252 NS = NS - 1: IF NS < 0 THEN NS = 0
20254  GOTO 20230
```

```
20260  IF NAME$(NS + 1) = "" THEN 20300
20270 NS = NS + 1: GOTO 20230
20300  PRINT : PRINT
20305  IF NT <  > 0 THEN  RETURN
20310  PRINT "DO YOU WANT TO SORT THESE NAMES
      (Y/N) ";: GET A$
20320  IF A$ = "Y" THEN 20340
20330  RETURN
20340  GOSUB 35000
20350  FOR I = 1 TO NS - 1: FOR J = 1 TO NS - 1
20360  IF NAME$(J) > NAME$(J + 1) THEN T$ = NA$(J):NA
     $(J) = NA$(J + 1):NA$(J + 1) = T$
20370  NEXT J,I
20380  RETURN
21000  REM !----------------------!
21010  REM !  ENTER NEW GRADES    !
21020  REM !----------------------!
21030  HOME
21035  IF NS = 0 THEN  PRINT "YOU CAN NOT ENTER GRADE
     S WITHOUT NAMES  FOR THE STUDENTS ON FILE": GET
     A$: RETURN
21040  PRINT "THERE ARE NOW "NT" TESTS. (MAX. ALLOWED
     =25)"
21050  PRINT "WHAT SET OF TESTS DO YOU WISH TO ENTER?
     "
21060  PRINT "ENTER A NUMBER BETWEEN 1 AND "NT + 1" "
     ;: INPUT A$
21070 A =  VAL (A$)
21080  IF A > (NT + 1) OR A < 1 THEN  RETURN
21082  PRINT : INPUT "WHAT DO YOU WISH TO CALL THIS T
     EST      (EXACTLY 3 NON-SPACE CHARATERS PLEASE)
      ";TH$(A)
21083  IF  LEN (TH$(A)) <  > 3 THEN 21082
21088  PRINT
21090  IF A = NT + 1 THEN NT = A: INPUT "WHAT IS THE
     WEIGHT OF THIS TEST ";PER(A)
21100  PRINT : PRINT
21102  PRINT "DO YOU WISH TO ENTER FROM THE CARD
      READER (R) OR FROM THE KEYBOARD (K) ";: GET A$
21104  IF A$ = "R" THEN 33000
21106  IF A$ = "K" THEN  PRINT : PRINT : GOTO 21110
21108  GOTO 21102
21110  PRINT : PRINT "DO YOU WISH TO USE THE HIDDEN N
     UMERIC   PAD  (Y/N) ";: GET A$: PRINT
21113  PRINT : PRINT : PRINT
21114  FOR I = 1 TO NS
21115  IF  LEFT$ (NAME$(I),1) = "*" THEN 21140
21120  PRINT I". "NAME$(I);: HTAB 20
21125  IF A$ = "Y" THEN  GOSUB 21200: GOTO 21131
21130  INPUT T$
21131  IF T$ <  > "<" THEN G%(I,A) =  VAL (T$): GOTO
     21140
21132 I = I - 1: IF I < 1 THEN I = 1
21133  PRINT : GOTO 21120
21140  NEXT I
21145 AG = 0
21150  RETURN
21200  REM NUMERIC PAD
21210 B = 0
21220  GET T$
21230  IF T$ = "<" THEN  PRINT "<": RETURN
21240  IF  ASC (T$) = 13 THEN T$ =  STR$ (B): PRINT :
      RETURN
```

```
21250  IF T$ = "7" THEN B = 10 * B + 7: PRINT "7";: GOTO
       21220
21260  IF T$ = "8" THEN B = 10 * B + 8: PRINT "8";: GOTO
       21220
21270  IF T$ = "9" THEN B = 10 * B + 9: PRINT "9";: GOTO
       21220
21280  IF T$ = "U" THEN B = 10 * B + 4: PRINT "4";: GOTO
       21220
21290  IF T$ = "I" THEN B = 10 * B + 5: PRINT "5";: GOTO
       21220
21300  IF T$ = "O" THEN B = 10 * B + 6: PRINT "6";: GOTO
       21220
21310  IF T$ = "J" THEN B = 10 * B + 1: PRINT "1";: GOTO
       21220
21320  IF T$ = "K" THEN B = 10 * B + 2: PRINT "2";: GOTO
       21220
21330  IF T$ = "L" THEN B = 10 * B + 3: PRINT "3";: GOTO
       21220
21340  IF T$ = " " THEN B = 10 * B: PRINT "0";: GOTO
       21220
21350  HTAB 20: PRINT "            "; CHR$ (7);: HTAB 2
       0: GOTO 21200
22000  REM !----------------------!
22010  REM !  MODIFY BY NAME OR # !
22020  REM !----------------------!
22030  HOME
22040  INPUT "WHAT IS THE NAME OR NUMBER OF THE
       STUDENT THAT YOU WISH TO MODIFY ";NM$
22050  IF  LEN (NM$) = 0 THEN  RETURN
22060  IF  VAL (NM$) <  > 0 THEN A =  VAL (NM$): GOTO
       23000
22070  GOSUB 16000
22080  IF SN = 0 THEN  RETURN
22090 A = SN
23000  REM !----------------------!
23010  REM !  MODIFY BY NUMBER    !
23020  REM !----------------------!
23050  IF A > NS OR A < 1 THEN  RETURN
23070 SN = A
23100  PRINT : PRINT
23110  PRINT "1. "NAME$(SN)
23120  FOR I = 1 TO NT
23130  PRINT (I + 1)". ("TH$(I)") "G%(SN,I)
23140  NEXT I
23150  PRINT : PRINT
23160  INPUT "WHICH FIELD ";A$
23170 A =  VAL (A$): IF A = 0 THEN  RETURN
23180  PRINT
23190  IF A > 1 THEN 23300
23200  INPUT "INPUT CORRECT NAME ";A$
23210  IF A$ = "" THEN  RETURN
23215 AG = 0
23220 NAME$(SN) = A$: HOME : GOTO 23110
23300  PRINT "ENTER NEW VALUE FOR "TH$(A - 1)" ";
23310  INPUT B$
23320  IF B$ = "" THEN  RETURN
23330 G%(SN,A - 1) =  VAL (B$)
23335 AG = 0
23340  HOME : GOTO 23100
24000  REM !----------------------!
24010  REM !    MODIFY WEIGHTS    !
24020  REM !----------------------!
24030  HOME
```

```
24040  PRINT "HERE ARE THE PRESENT WEIGHTS": PRINT
24050  FOR I = 1 TO NT
24060  PRINT I". ("TH$(I)") "PER(I)
24070  NEXT I
24080  PRINT : PRINT
24090  INPUT "WHICH FIELD TO MODIFY ";A$
24100 A =  VAL (A$)
24110  IF A = 0 THEN  RETURN
24120  IF A ) NT THEN  RETURN
24130  PRINT "ENTER THE NEW VALUE FOR "TH$(A)" ";: INPUT
     B$
24140 B =  VAL (B$)
24150  IF B$ = "" THEN  RETURN
24160 PER(A) = B
24165 AG = 0
24170  GOTO 24030
25000  REM !------------------------!
25010  REM !     ENTER FROM DISK      !
25020  REM !------------------------!
25030  HOME
25035  ONERR  GOTO 25300
25040  INPUT "ENTER NAME OF FILE TO BE LOADED
     ";FL$
25050  IF FL$ = "" THEN  RETURN
25060 D$ =  CHR$ (4)
25070  PRINT D$"OPEN "FL$
25080  PRINT D$"READ "FL$
25090  INPUT NS
25100  FOR I = 1 TO NS
25110  INPUT NAME$(I)
25120  NEXT I
25130  INPUT NT
25140  FOR TN = 1 TO NT
25150  INPUT PER(TN)
25155  INPUT TH$(TN)
25160  FOR SN = 1 TO NS
25170  INPUT G%(SN,TN)
25180  NEXT SN
25190  NEXT TN
25200  PRINT D$"CLOSE"
25205 AG = 0
25210  RETURN
25300  CALL 768: PRINT "THAT FILE WILL NOT LOAD (RTN
     FOR MENU)": PRINT : GOTO 25040
26000  REM !-----------------------!
26010  REM !     SAVE TO DISK       !
26020  REM !-----------------------!
26025  ONERR  GOTO 26300
26030  HOME
26032  IF NS = 0 THEN  PRINT "THERE IS NO DATA ON FIL
     E": GET A$: RETURN
26034  FLASH : PRINT "WARNING--SAVING DATA WILL REPLA
     CE FILE ": NORMAL : PRINT
26040  PRINT "READY TO SAVE DATA TO DISK": PRINT
26050  IF  LEN (FL$) <  > 0 THEN  PRINT "USE "FL$" AS
      FILE NAME (Y/N) ": GET A$: IF A$ = "Y" THEN 261
     00
26055  PRINT
26060  INPUT "WHAT FILE NAME ";FL$
26070  IF FL$ = "" THEN  RETURN
26100 D$ =  CHR$ (13) +  CHR$ (4)
26110  PRINT D$"OPEN "FL$
26120  PRINT D$"WRITE "FL$
```

```
26130  PRINT NS
26140  FOR I = 1 TO NS
26150  PRINT NAME$(I)
26160  NEXT I
26170  PRINT NT
26180  FOR TN = 1 TO NT
26190  PRINT PER(TN)
26195  PRINT TH$(TN)
26200  FOR SN = 1 TO NS
26210  PRINT G%(SN,TN)
26220  NEXT SN
26230  NEXT TN
26235  PRINT D$"CLOSE"
26240  RETURN
26300  CALL 768
26301  IF  PEEK (222) = 4 THEN  PRINT "YOUR DISKETTE
     IS WRITE PROTECTED": GOTO 26311
26302  IF  PEEK (222) = 10 THEN  PRINT "THAT FILE IS
     LOCKED": GOTO 26311
26303  IF  PEEK (222) = 9 THEN  PRINT "YOUR DISKETTE
     IS FULL": PRINT "SAVE FILE ON ANOTHER DISKETTE "
     : GOTO 26060
26310  PRINT "THAT FILE NAME WILL NOT WORK - HIT
      RETURN IF YOU WISH TO RETURN TO THE      MENU TO
      CHECK THE CATALOG": GOTO 26060
26311  PRINT "(JUST RETURN FOR MENU)": GOTO 26060
27000  REM !-----------------------!
27010  REM !  STATISTICAL ANALYSIS !
27020  REM !-----------------------!
27030  HOME
27040  PRINT "YOU MANY GET A STATISTICAL ANALYSIS FOR
      ANY TEST OR FOR THE PRESENT AVERAGES": PRINT
27060  INPUT "ENTER EITHER A TEST NUMBER OR AN 'A' FO
     RAVERAGES ";A$
27070  IF A$ = "A" THEN  GOTO 27500
27080 TN =  VAL (A$)
27100  REM !-----------------------!
27110  REM !     ANALYZE A TEST      !
27120  REM !-----------------------!
27130  HOME
27152 NA = 0:NB = 0:NC = 0:ND = 0:TA = 0:TV = 0
27153 N50 = 0:N40 = 0:N30 = 0:N20 = 0:N10 = 0:N0 = 0
27154 LG = 100:HG = 0
27155  IF TN < 1 OR TN > NT THEN  RETURN
27156  GOSUB 35000
27160  FOR SN = 1 TO NS
27162  IF  LEFT$ (NAME$(SN),1) = "*" OR G%(SN,TN) < 0
      THEN TV = TV + 1: GOTO 27230
27170  IF G%(SN,TN) >  = 90 THEN NA = NA + 1: GOTO 27
     220
27180  IF G%(SN,TN) >  = 80 THEN NB = NB + 1: GOTO 27
     220
27190  IF G%(SN,TN) >  = 70 THEN NC = NC + 1: GOTO 27
     220
27200  IF G%(SN,TN) >  = 60 THEN ND = ND + 1: GOTO 27
     220
27201  IF G%(SN,TN) >  = 50 THEN N50 = N50 + 1: GOTO
     27220
27202  IF G%(SN,TN) >  = 40 THEN N40 = N40 + 1: GOTO
     27220
27203  IF G%(SN,TN) >  = 30 THEN N30 = N30 + 1: GOTO
     27220
27204  IF G%(SN,TN) >  = 20 THEN N20 = N20 + 1: GOTO
     27220
```

```
27205  IF G%(SN,TN) >  = 10 THEN N10 = N10 + 1: GOTO
     27220
27206 N0 = N0 + 1
27220 TA = TA + G%(SN,TN)
27222  IF G%(SN,TN) > HG THEN HG = G%(SN,TN)
27224  IF G%(SN,TN) >  = 0 AND G%(SN,TN) < LG THEN LG
     = G%(SN,TN)
27230  NEXT SN
27240 TA = TA / (NS - TV)
27245  IF RF = 1 THEN RF = 0: HOME : RETURN
27250  IF PF$ = "ON" THEN  GOSUB 18000
27260  HOME
27265  PRINT "TEST # "TN"        "FL$
27280  PRINT "THE AVERAGE ON TEST #"TN" IS " INT (TA +
     .5)
27290  PRINT
27300  PRINT "90";:I = NA: GOSUB 27400
27310  PRINT "80";:I = NB: GOSUB 27400
27320  PRINT "70";:I = NC: GOSUB 27400
27330  PRINT "60";:I = ND: GOSUB 27400
27340  PRINT "50";:I = N50: GOSUB 27400
27341  PRINT "40";:I = N40: GOSUB 27400
27342  PRINT "30";:I = N30: GOSUB 27400
27343  PRINT "20";:I = N20: GOSUB 27400
27344  PRINT "10";:I = N10: GOSUB 27400
27345  PRINT "<9";:I = N0: GOSUB 27400
27350  PRINT : PRINT "THE LOWEST GRADE IS "LG
27360  PRINT "THE HIGHEST GRADE IS "HG
27370  PRINT : PRINT  CHR$ (4)"PR#0"
27380  IF PF$ = "OFF" THEN  GET A$
27390  RETURN
27400  REM !-------------------------!
27405  REM !     PRINT I *'S         !
27410  REM !-------------------------!
27420  IF I = 0 THEN  PRINT : RETURN
27425  PRINT " ";
27430  FOR J = 1 TO I
27440  PRINT "*";
27450  NEXT J
27460  PRINT " ("I")"
27470  RETURN
27500  REM !-----------------------!
27510  REM !     ANALYZE AVERAGES  !
27520  REM !-----------------------!
27530  HOME
27535  GOSUB 35000
27540  IF AG = 1 THEN 27560
27550  FOR SN = 1 TO NS: GOSUB 11000: NEXT SN:AG = 1
27560 NA = 0:NB = 0:NC = 0:ND = 0:TA = 0
27565 N50 = 0:N40 = 0:N30 = 0:N20 = 0:N10 = 0:N0 = 0
27570 LG = 100:HG = 0:TV = 0
27580  FOR SN = 1 TO NS
27585  IF  LEFT$ (NAME$(SN),1) = "*" THEN TV = TV + 1
     : GOTO 27670
27590  IF AVE(SN) >  = 90 THEN NA = NA + 1: GOTO 2764
     0
27600  IF AVE(SN) >  = 80 THEN NB = NB + 1: GOTO 2764
     0
27610  IF AVE(SN) >  = 70 THEN NC = NC + 1: GOTO 2764
     0
27620  IF AVE(SN) >  = 60 THEN ND = ND + 1: GOTO 2764
     0
```

```
27621  IF AVE(SN) >  = 50 THEN N50 = N50 + 1: GOTO 27
     640
27622  IF AVE(SN) >  = 40 THEN N40 = N40 + 1: GOTO 27
     640
27623  IF AVE(SN) >  = 30 THEN N30 = N30 + 1: GOTO 27
     640
27624  IF AVE(SN) >  = 20 THEN N20 = N20 + 1: GOTO 27
     640
27625  IF AVE(SN) >  = 10 THEN N10 = N10 + 1: GOTO 27
     640
27630 N0 = N0 + 1
27640 TA = TA + AVE(SN)
27650  IF AVE(SN) > HG THEN HG = AVE(SN)
27660  IF AVE(SN) < LG THEN LG = AVE(SN)
27670  NEXT SN
27680 TA = TA / (NS - TV)
27690  IF PF$ = "ON" THEN  GOSUB 18000
27700  HOME
27705  PRINT "FINAL AVERAGES FOR "FL$
27710  PRINT "THE AVERAGE FINAL GRADE IS " INT (TA +
    .5)
27720  GOTO 27290
28000  REM !--------------------------!
28010  REM !       CURVE   GRADES      !
28020  REM !--------------------------!
28025  DEF  FN CURVE(G) = (G * A) + B
28030  HOME
28040  PRINT "WHICH TEST DO YOU WISH TO CURVE?
      ENTER A NUMBER BETWEEN 1 AND "NT" ";
28050  INPUT "";A$:TN =  VAL (A$)
28055  IF TN < 1 OR TN > NT THEN  RETURN
28056  HOME : PRINT "AUTOMATIC OR MANUAL (A/M) ";: GET
    A$
28057  IF A$ = "A" THEN 31000
28058  IF A$ = "M" THEN 28060
28059  RETURN
28060 RF = 1: GOSUB 27152: HOME
28070  PRINT "WE ARE READY TO CURVE TEST #"TN
28072  PRINT "PRESENTLY, THE STATISTICS ARE"
28073  PRINT "THE AVERAGE IS " INT (TA + .5)
28074  PRINT "THE LOW GRADE IS "LG
28075  PRINT "THE HIGH GRADE IS "HG
28076  PRINT : PRINT
28080  PRINT : PRINT "THIS IS THE CURVING FUNCTION"
28090  PRINT "       NEW GRADE=(A*G)+B"
28100  PRINT : PRINT
28110  PRINT "G = THE GRADES TO BE CURVED"
28120  PRINT "A = THE MULTIPLIER"
28130  PRINT "B = THE ADDITIONAL POINTS"
28140  PRINT : PRINT
28142  PRINT "DO YOU STILL WISH TO CURVE (Y/N) ";: GET
    A$
28144  IF A$ <  > "Y" THEN  RETURN
28146  PRINT : PRINT
28148 AG = 0: REM  NEED TO RE-AVERAGE
28150  INPUT "VALUE TO USE FOR 'A' ";A
28160  INPUT "VALUE TO USE FOR 'B' ";B
28165  GOSUB 35000
28170  FOR SN = 1 TO NS
28175  IF G%(SN,TN) < 0 THEN 28200
28180 G = G%(SN,TN)
28190 G%(SN,TN) =  INT ( FN CURVE(G) + .5)
```

```
28200  NEXT SN
28210  RETURN
29000  REM !----------------------!
29005  REM !  SORT FOR DROP       !
29010  REM !----------------------!
29015  HOME
29016  IF NT < 2 THEN  PRINT "YOU CANNOT ORDER TESTS
     WHEN YOU HAVE    LESS THAN 2.": GET A$: RETURN
29020  PRINT "THIS MODULE ALLOWS YOU TO SORT SEVERAL
      GRADES SO THAT YOU CAN MODIFY WEIGHTS   TO DROP
      THE LOWEST ONES"
29025  PRINT : PRINT
29030  PRINT "ENTER THE TEST NUMBER OF EACH TEST TO B
     EINCLUDED (RETURN WHEN DONE)"
29100 TS = 0
29200  INPUT A$
29300  IF A$ = "" THEN 29600
29400 TS(TS + 1) =  VAL (A$)
29500 TS = TS + 1
29550  GOTO 29200
29600  IF TS = 0 THEN  RETURN
29602  GOSUB 35000
29605  FOR SN = 1 TO NS
29610  FOR J = 1 TO TS - 1
29620  FOR I = 1 TO TS - 1
29624  IF G%(SN,TS(I + 1)) < 0 THEN 29670
29625  IF G%(SN,TS(I)) < 0 THEN 29640
29630  IF G%(SN,TS(I)) <  = G%(SN,TS(I + 1)) THEN 296
     70
29640 T = G%(SN,TS(I))
29650 G%(SN,TS(I)) = G%(SN,TS(I + 1))
29660 G%(SN,TS(I + 1)) = T
29670  NEXT I
29675  PRINT ".";
29680  NEXT J
29685  PRINT SN;
29690  NEXT SN
29695 AG = 0
29700  RETURN
30000  REM !----------------------!
30005  REM ! REPLACE LOWEST GRADE!
30010  REM !----------------------!
30015  HOME
30016  IF NT < 2 THEN  PRINT "YOU CANNOT REPLACE WHEN
      YOU HAVE LESS   THAN 2 TESTS": GET A$: RETURN
30020  PRINT "THIS MODULE WILL REPLACE THE LOWEST
      GRADE OF SEVERAL TESTS WITH A TEST OF   YOUR CH
     OICE - IF IT IS HIGHER"
30025  PRINT : PRINT
30030  PRINT "INPUT THE TESTS TO BE CONSIDERED
      (RETURN WHEN DONE)"
30120 TS = 0
30130  INPUT A$
30140  IF A$ = "" THEN 30180
30150 TS(TS + 1) =  VAL (A$)
30160 TS = TS + 1
30170  GOTO 30130
30180  IF TS = 0 THEN  RETURN
30190  INPUT "WHAT TEST NUMBER SHOULD BE USED AS THE
      REPLACEMENT ";TR
30195  GOSUB 35000
30200  FOR SN = 1 TO NS
30210 TL = 1
```

```
30220  FOR I = 1 TO TS
30225  IF G%(SN,TS(TL)) < 0 THEN TL = I
30230  IF G%(SN,TS(TL)) > G%(SN,TS(I)) AND G%(SN,TS(I
    )) > = 0 THEN TL = I
30240  NEXT I
30250  IF G%(SN,TS(TL)) < G%(SN,TR) AND G%(SN,TS(TL))
    > = 0 THEN G%(SN,TS(TL)) = G%(SN,TR)
30260  NEXT SN
30265 AG = 0
30270  RETURN
31000  REM !----------------------!
31010  REM ! AUTOMATIC CURVING    !
31020  REM !----------------------!
31030  HOME
31035 RF = 1: GOSUB 27152
31040  PRINT "THE GRADES WILL AUTOMATICALLY BE CURVED
    FOR TEST "TN: PRINT
31041  PRINT "THE PRESENT STATISTICS ARE"
31042  PRINT "THE AVERAGE IS " INT (TA + .5)
31043  PRINT "THE LOW GRADE IS "LG
31044  PRINT "THE HIGH GRADE IS "HG
31045  PRINT
31050  INPUT "WHAT DO YOU WANT THE NEW HIGH SCORE TO
    BE ";R1
31060  PRINT : PRINT "YOU MAY NOW ENTER EITHER THE NE
    W LOW     SCORE OR THE NEW AVERAGE"
31065  PRINT
31070  PRINT "WHICH DO YOU PREFER (L/A) ";: GET A$
31075  PRINT
31080  IF A$ = "A" THEN  INPUT "NEW AVERAGE=";R2: GOTO
    31100
31090  IF A$ = "L" THEN  INPUT "NEW LOW SCORE=";R2: GOTO
    31200
31095  RETURN
31100  REM  NEW AVERAGE
31105  IF HG - TA = 0 THEN  PRINT "YOU CANNOT CURVE T
    HE GRADES AUTO-        MATICALLY IF THEY ARE ALL
    THE SAME.      (RTN FOR MENU)": GET A$: RETURN
31120 A = (R1 - R2) / (HG - TA)
31130  GOTO 31300
31200  REM  NEW LOW GRADE
31210  IF HG - LG = 0 THEN HG = TA: GOTO 31105
31220 A = (R1 - R2) / (HG - LG)
31300  REM  FIND 'B' THE SAME FOR BOTH
31310 B = R1 - (A * HG)
31311  HOME : PRINT "THE CALCULATED VALUES FOR CURVIN
    G ARE"
31312  PRINT "     A="A: PRINT "     B="B: PRINT
31313  PRINT "PRESS N TO ABORT, ANY OTHER KEY TO
    CONTINUE";: GET X$
31314  IF X$ = "N" THEN  RETURN
31320  GOTO 28165: REM  USE FORMULA NOW THAT A&B HAVE
    BEEN FOUND
32000  REM !----------------------!
32005  REM !   REPORT FORMAT      !
32010  REM !----------------------!
32015  IF PF$ = "OFF" THEN  RETURN
32020  HOME
32021  PRINT
32025  PRINT  CHR$ (4)"PR#0"
32030  PRINT "INPUT YOUR HEADING BELOW": PRINT
32040  INPUT HD$
32045 HD$ = HD$ + "        " + FL$
```

```
32050  PRINT : PRINT
32060  INPUT "HOW MANY SPACES BETWEEN LINES ";SP$
32062  PRINT : INPUT "HOW MANY NAMES DO YOU WANT ON E
    ACH       PAGE ";A$
32063  IF  LEN (A$) = 0 THEN LP = 125: GOTO 32065
32064 LP =  VAL (A$)
32065  IF PF$ = "ON" THEN  GOSUB 18000
32070  PRINT HD$
32075  PRINT
32080  RETURN
33000  REM !-----------------------!
33002  REM ! INPUT FROM CARDREADER !
33004  REM !-----------------------!
33005  IF  PEEK (50176) = 138 AND  PEEK (50177) = 72 THEN
    33009
33006  HOME : PRINT "A PROPER CARD READER MUST BE IN
    SLOT 4. LINE 33005 OF THE GRADEBOOK PROGRAM HAS
    DETERMINED THAT THE READER IS NOT THERE.";
33007  PRINT "REFER TO CHAPTERS 8 AND 9 FOR MORE
    DETAILS. (RTN FOR MENU)"
33008 NT = NT - 1: GET A$: RETURN
33009 AG = 0:NH = 0:NE = 0:RQ = 0
33010  HOME
33020  PRINT "MAKE SURE THE CARD READER IS TURNED ON
    AND THAT THE CARDS (IN THE PROPER ORDER)ARE IN
    THE HOPPER": PRINT
33030  INPUT "HOW MANY QUESTIONS ARE ON THIS TEST
    (50 MAX) ";NQ
33035  PRINT : INPUT "IF THERE ARE ANY PARTIAL CREDIT
    PROBLEMSON THIS TEST WHAT IS THEIR RELATIVE
    WORTH ";A$
33036 PC =  VAL (A$): IF PC = 0 THEN PC = 1
33038  PRINT : IF PF$ = "ON" THEN  PRINT "MAKE SURE T
    HE PRINTER IS READY FOR THE  INDIVIDUAL GRADES":

33040  PRINT : INPUT "HIT RETURN TO START READING ";A
    $
33045  IF PF$ = "ON" THEN  GOSUB 18000
33050  FOR I = 1 TO NS:G%(I,A) =  - 1: NEXT I
33055  REM  INITIALIZE TEST SCORES
33056 PT = 0: REM  PEOPLE TAKING TEST
33060  FOR I = 1 TO 50: FOR J = 1 TO 5:K%(I,J) = 0: NEXT
    J,I
33061  REM
33070  REM  NOW READ A CARD
33080  CALL  - 15168
33090  POKE  - 16192,4
33100  CALL  - 15360:R$ = "": FOR I = 520 TO 579: IF
    PEEK (I) > 128 THEN R$ = R$ +  CHR$ ( PEEK (I) -
    128)
33101  NEXT I
33102 SN =  VAL (R$):NL = 0:WW = 0: IF SN = 88888 OR
    SN = 99999 THEN 33110
33103  IF SN >  = 1 AND SN <  = NS THEN 33110
33104  PRINT  CHR$ (4)"PR#0"
33105  INPUT "WHAT IS THIS STUDENT'S NUMBER ";SN
33106  IF PF$ = "ON" THEN  GOSUB 18000:NL = NL + 1
33107  GOTO 33103
33110  IF  LEFT$ (R$,5) = "88888" THEN 33500: REM LAS
    T CARD
33115  IF  LEFT$ (R$,5) <  > "99999" THEN PT = PT + 1
    : GOTO 33130
```

```
33120 K$ = R$:NP = NQ: FOR I = 6 TO NQ + 5: IF  MID$
     (K$,I,1) <  > "*" THEN 33128
33122 NP = NP + PC - 1
33128 RQ = RQ + 1: NEXT I: GOTO 33061
33130 NM = 0
33131  PRINT SN"."NAME$(SN):NL = 1
33132  FOR I = 6 TO NQ + 5
33133 T$ =  MID$ (R$,I,1)
33134  IF T$ = "A" THEN K%(I - 5,1) = K%(I - 5,1) + 1

33135  IF T$ = "B" THEN K%(I - 5,2) = K%(I - 5,2) + 1

33136  IF T$ = "C" THEN K%(I - 5,3) = K%(I - 5,3) + 1

33137  IF T$ = "D" THEN K%(I - 5,4) = K%(I - 5,4) + 1

33138  IF T$ = "E" THEN K%(I - 5,5) = K%(I - 5,5) + 1

33139  IF T$ <  > "*" THEN 33144
33140  PRINT : PRINT  CHR$ (4)"PR#0":NL = NL + 1
33141  PRINT "WHAT IS THE ANSWER TO QUESTION #"I - 5"
            (A,B,C,D,E,-) ";: GET T$: IF  NOT (T$ =
     "A" OR T$ = "B" OR T$ = "C" OR T$ = "D" OR T$ =
     "E" OR T$ = "-") THEN 33141
33142  IF PF$ = "ON" THEN  GOSUB 18000
33143  GOTO 33134
33144  IF  MID$ (K$,I,1) = "-" THEN 33160
33145  IF  MID$ (K$,I,1) = "*" THEN 33180
33150  IF  MID$ (K$,I,1) <  > T$ THEN NM = NM + 1: PRINT
     I - 5" ";:WW = WW + 1
33152  IF WW > 10 THEN  PRINT :NL = NL + 1:WW = 0
33160  NEXT I
33162 G%(SN,A) =  INT ((100 - ((NM * 100) / NP)) + .5
     ): PRINT : PRINT "GRADE = "G%(SN,A):NL = NL + 1
33164  IF  NOT ((NL = 5) OR (NL = 11)) THEN  PRINT :N
     L = NL + 1: GOTO 33164
33170  GOTO 33061
33180  REM !-------------------------!
33182  REM ! PARTIAL CREDIT QUESTION !
33184  REM !-------------------------!
33190 NM = NM + PC
33200  IF  MID$ (R$,I,1) = "A" THEN NM = NM - PC: GOTO
     33160
33205 WW = WW + 2
33210  IF  MID$ (R$,I,1) = "B" THEN NM = NM - .75 * P
     C: PRINT I - 5".75 ";: GOTO 33160
33220  IF  MID$ (R$,I,1) = "C" THEN NM = NM - .5 * PC
     : PRINT I - 5".5 ";: GOTO 33160
33230  IF  MID$ (R$,I,1) = "D" THEN NM = NM - .25 * P
     C: PRINT I - 5".25 ";: GOTO 33160
33235  PRINT I - 5;" ";
33240  GOTO 33160
33500  REM !-----------------------!
33502  REM !  PRINT TEST RESULTS   !
33503  REM !-----------------------!
33505  PRINT  CHR$ (4)"PR#0": POKE  - 16192,0: REM  R
     EADER OFF
33506  IF PF$ = "ON" THEN  INPUT "PREPARE PRINTER FOR
      ANALYSIS - HIT       RETURN WHEN READY ";A$
33507 TN = A: GOSUB 27152: PRINT
33540  IF PF$ = "ON" THEN  GOSUB 18000
33550  PRINT "ANALYSIS FOR TEST "A" - "FL$
```

```
33560  PRINT : PRINT
33570  PRINT "QUES.   A   B   C   D   E   CORRECT   %RIGHT"
33580  PRINT
33590  FOR I = 1 TO NQ
33595  IF I < 10 THEN  PRINT " ";
33596  PRINT I"    ";
33600  FOR J = 1 TO 5
33602  IF K%(I,J) < 10 THEN  PRINT " ";
33604  PRINT " "K%(I,J);
33606  NEXT J
33607  GOSUB 37000
33608  PRINT
33700  IF I = 10 OR I = 20 OR I = 30 OR I = 40 THEN  PRINT

33710  NEXT I
33711  PRINT : PRINT : PRINT  INT (((NE / RQ) * 100) +
     .5);"% OF THE QUESTIONS WERE EASY"
33712  PRINT  INT (((NH / RQ) * 100) + .5);"% OF THE
    QUESTIONS WERE HARD"
33718  PRINT : PRINT  CHR$ (4)"PR#0"
33719  IF PF$ = "OFF" THEN  GET A$
33720  RETURN
35000  REM ------FLASH WORKING-------
35010  HOME
35020  VTAB 12: HTAB 15
35030  FLASH : PRINT "W O R K I N G": NORMAL
35040  RETURN
36000  REM ----PRINT TEST TITLES------
36005  IF NT = 0 THEN  RETURN
36010  PRINT "NUM. NAME              ";
36015  IF ALL = 0 THEN 36050
36020  FOR I = 1 TO NT
36030  PRINT " ";: PRINT TH$(I);
36040  NEXT I
36050  PRINT " ";: INVERSE : PRINT "AVE": NORMAL : PRINT
     : RETURN
36302  IF  PEEK (222) = 10 THEN  PRINT "THAT FILE IS
    LOCKED": GOTO 26311
37000  REM !------------------!
37005  REM ! ANSWER  ANALYSIS !
37010  REM !------------------!
37020 T$ =  MID$ (K$,I + 5,1)
37030  PRINT "     "T$"    ";
37040  IF T$ = "-" THEN  PRINT "NO RIGHT ANSWER";: GOTO
     37100
37050  IF T$ = "*" THEN  PRINT "NOT MULTIPLE CHOICE";
     : GOTO 37100
37060 CR =  INT ((100 * (K%(I,( ASC (T$) - 64)) / PT)
     ) + .5)
37070  IF CR < 100 THEN  PRINT " ";: IF CR < 10 THEN
      PRINT " ";
37075  PRINT "  "CR"  ";
37080  IF CR >  = 80 THEN  PRINT "EASY";:NE = NE + 1
37085  IF CR <  = 20 THEN  PRINT "HARD";:NH = NH + 1
37100  RETURN
```

Appendix B

FLEXIBLE SORT

```
10   REM !--------------------!
11   REM ! FLEXIBLE SORT DEMO !
12   REM !BY  JOHN BLANKENSHIP!
13   REM !  FOR USE WITH THE  !
14   REM ! COMPLETE GRADEBOOK !
15   REM !       SYSTEM       !
16   REM !  COPYRIGHT   9/83  !
17   REM !--------------------!
99   GOSUB 16500: REM  INITIALIZE
100   HOME
110   POKE 216,0: REM  RESET ERROR FLAG
115   PRINT "FLEXIBLE SORT DEMO": PRINT : PRINT
120   PRINT "L-LOAD FILE"
125   PRINT "S-SAVE FILE"
130   PRINT "O-ORDER THE FILE"
135   PRINT "E-END"
140   PRINT : PRINT
145   PRINT "ENTER LETTER OF YOUR CHOICE ";: GET A$
150   IF A$ = "L" THEN  GOSUB 25000: GOTO 100
155   IF A$ = "S" THEN  GOSUB 26000: GOTO 100
160   IF A$ = "O" THEN  GOSUB 500: GOTO 100
165   IF A$ = "E" THEN  END
170   PRINT  CHR$ (7): GOTO 100
500   REM !------------------!
501   REM ! INPUT PARAMETERS !
502   REM !      FOR SORT     !
503   REM !------------------!
510   HOME
515   PRINT "YOU MAY SORT THE FILE ON ANY OF THE      F
      OLLOWING": PRINT : PRINT
520   PRINT "N-NAMES"
525   PRINT "T-ANY TEST"
530   PRINT "A-FINAL AVERAGES"
```

```
535  PRINT : PRINT
540  PRINT "ENTER LETTER OF YOUR CHOICE ";: GET A$
545  IF A$ = "N" THEN  GOSUB 1000: RETURN
550  IF A$ = "A" THEN  GOSUB 3000: RETURN
555  IF A$ <  > "T" THEN  RETURN
560  PRINT : PRINT
565  PRINT "THERE ARE NOW "NT" TESTS"
570  INPUT "WHICH ONE WOULD YOU LIKE TO USE AS THE  K
     EY FOR THE SORT ";A$
575  IF  VAL (A$) = 0 THEN  RETURN
580 K =  VAL (A$)
585  GOSUB 2000
590  RETURN
900  REM !-----------------------!
901  REM ! EXCHANGE ALL ITEMS FOR !
902  REM ! STUDENT #'S SN & SN+1  !
903  REM !-----------------------!
905 T$ = NAME$(SN):NAME$(SN) = NAME$(SN + 1):NAME$(SN
      + 1) = T$
910  FOR TN = 1 TO NT
920 T = G%(SN,TN):G%(SN,TN) = G%(SN + 1,TN):G%(SN + 1
     ,TN) = T
930  NEXT TN
935 T = AVE(SN):AVE(SN) = AVE(SN + 1):AVE(SN + 1) = T

940  PRINT ".";
950  RETURN
1000  REM !--------------!
1001  REM ! SORT BY NAME !
1002  REM !--------------!
1003  GOSUB 4000
1005  FOR I = 1 TO NS - 1
1010  FOR SN = 1 TO NS - 1
1020  IF NAME$(SN) > NAME$(SN + 1) THEN  GOSUB 900
1030  NEXT SN,I
1040  RETURN
2000  REM !-----------------!
2001  REM ! SORT BY TEST # K!
2002  REM !-----------------!
2003  GOSUB 4000
2005  FOR I = 1 TO NS - 1
2010  FOR SN = 1 TO NS - 1
2020  IF G%(SN,K) < G%(SN + 1,K) THEN  GOSUB 900
2030  NEXT SN,I
2040  RETURN
3000  REM !------------------!
3001  REM ! SORT BY AVERAGES !
3002  REM !------------------!
3003  GOSUB 4000
3005  REM  FIRST AVERAGE GRADES
3010  FOR SN = 1 TO NS
3020  GOSUB 11000
3030  NEXT SN
3040  REM  NOW SORT
3050  FOR I = 1 TO NS - 1
3060  FOR SN = 1 TO NS - 1
3070  IF AVE(SN) < AVE(SN + 1) THEN  GOSUB 900
3080  NEXT SN,I
3090  RETURN
4000  REM !---------!
4001  REM ! MESSAGE !
4002  REM !---------!
4005  HOME
```

```
4010  PRINT "I ASSUME THAT YOU WILL ONLY SORT GRADES
      OCCASIONALLY, SO I USED A BUBBLE SORT INBOTH THI
      S DEMO AND THE GRADEBOOK PROGRAM"
4015  PRINT "A BUBBLE SORT IS EASY TO UNDERSTAND AND
      MODIFY, BUT IS IS VERY SLOW": PRINT
4020  PRINT "IF YOU HAVE A LARGE FILE JUST RELAX AND
      WAIT A WHILE"
4030  RETURN
11000  REM !------------------------!
11002  REM ! AVE GRADES FOR STUD SN !
11004  REM !------------------------!
11005 TP = 0: FOR TN = 1 TO NT
11006  IF G%(SN,TN) < 0 THEN 11008
11007 TP = TP + PER(TN)
11008  NEXT TN
11010 AVE = 0
11020  FOR TN = 1 TO NT
11025  IF G%(SN,TN) < 0 THEN 11040
11030 AVE = AVE + (PER(TN) / TP) * G%(SN,TN)
11040  NEXT TN
11050 AVE(SN) =  INT (AVE + .5)
11060  RETURN
16500  REM !----------------!
16505  REM ! INITIALIZATION !
16510  REM !----------------!
16520  DIM G%(125,25),NAME$(125),PER(25),AVE(125),TS(
      25),K%(50,5),TH$(25)
16530 PF$ = "OFF"
16535  TEXT
16539  REM  ONERR ROUTINE
16540  FOR I = 1 TO 10
16550  READ A
16560  POKE 767 + I,A
16570  NEXT I
16580  RETURN
16600  DATA  104,168,104,166,223,154,72,152,72,96
25000  REM !------------------------!
25010  REM !     ENTER FROM DISK      !
25020  REM !------------------------!
25030  HOME
25035  ONERR  GOTO 25300
25040  INPUT "ENTER NAME OF FILE TO BE LOADED
      ";FL$
25050  IF FL$ = "" THEN  RETURN
25060 D$ =  CHR$ (4)
25070  PRINT D$"OPEN "FL$
25080  PRINT D$"READ "FL$
25090  INPUT NS
25100  FOR I = 1 TO NS
25110  INPUT NAME$(I)
25120  NEXT I
25130  INPUT NT
25140  FOR TN = 1 TO NT
25150  INPUT PER(TN)
25155  INPUT TH$(TN)
25160  FOR SN = 1 TO NS
25170  INPUT G%(SN,TN)
25180  NEXT SN
25190  NEXT TN
25200  PRINT D$"CLOSE"
25205 AG = 0
25210  RETURN
```

```
25300  CALL 768: PRINT "THAT NAME WAS NOT FOUND (RTN
     FOR MENU)": PRINT : GOTO 25040
26000  REM !----------------------!
26010  REM !     SAVE TO DISK      !
26020  REM !----------------------!
26025  ONERR  GOTO 26300
26030  HOME
26032  IF NS = 0 THEN  PRINT "THERE IS NO DATA ON FIL
     E": GET A$: RETURN
26034  FLASH : PRINT "WARNING--SAVING DATA WILL REPLA
     CE FILE ": NORMAL : PRINT
26040  PRINT "READY TO SAVE DATA TO DISK": PRINT
26050  IF  LEN (FL$) <  > 0 THEN  PRINT "USE "FL$" AS
      FILE NAME (Y/N) ": GET A$: IF A$ = "Y" THEN 261
     00
26055  PRINT
26060  INPUT "WHAT FILE NAME ";FL$
26070  IF FL$ = "" THEN  RETURN
26100 D$ =  CHR$ (13) +  CHR$ (4)
26110  PRINT D$"OPEN "FL$
26120  PRINT D$"WRITE "FL$
26130  PRINT NS
26140  FOR I = 1 TO NS
26150  PRINT NAME$(I)
26160  NEXT I
26170  PRINT NT
26180  FOR TN = 1 TO NT
26190  PRINT PER(TN)
26195  PRINT TH$(TN)
26200  FOR SN = 1 TO NS
26210  PRINT G%(SN,TN)
26220  NEXT SN
26230  NEXT TN
26235  PRINT D$"CLOSE"
26240  RETURN
26300  CALL 768
26301  IF  PEEK (222) = 4 THEN  PRINT "YOUR DISKETTE
     IS WRITE PROTECTED": GOTO 26311
26302  IF  PEEK (222) = 10 THEN  PRINT "THAT FILE IS
     LOCKED": GOTO 26311
26303  IF  PEEK (222) = 9 THEN  PRINT "YOUR DISKETTE
     IS FULL": PRINT "SAVE FILE ON ANOTHER DISKETTE "
     : GOTO 26060
26310  PRINT "THAT FILE NAME WILL NOT WORK - HIT
      RETURN IF YOU WISH TO RETURN TO THE      MENU TO
      CHECK THE CATALOG": GOTO 26060
26311  PRINT "(JUST RETURN FOR MENU)": GOTO 26060
```

THE GRADEBOOK SYSTEM (APPLE II/IIe VERSION)
by John Blankenship
ISBN 0-13-362534-6
© 1984 by Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632
For use on the Apple II and Apple IIe

Dear Educator,

A mistake in the Gradebook Program was found shortly after the diskettes were produced. The problem occurs when grades are curved with the automatic option. Although the grades will be changed, the averages for each student will not. If the grades are saved to the disk, and then re-loaded, then the averages will be correct. You can modify the program so that averages will be correct immediately after curving, by performing the steps listed below:

1. RUN the Gradebook Program as described in the manual.
2. Use the E option and end the program.
3. Type in the following line: 28035 AG=0
4. Save the changed program by typing:

SAVE GRADEBOOK

DISK INCLUDED

THE GRADEBOOK SYSTEM

APPLE II/IIe

JOHN BLANKENSHIP

These chapter headings will give you an idea of the range of information that **John Blankenship** includes in this easy-to-use book:

An Introduction. Entering Names. Disk Operations. Entering Grades. Examining Grades. Making Modifications. Analyzing and Adjusting Grades. Using the Card Reader. Module Descriptions. Modifying Data Files. A Final Word. Appendix A GRADEBOOK. Appendix B FLEXIBLE SORT.

This book (it's not just a book—it's software too!) is meant to relieve you of tedious, time-consuming record-keeping responsibilities. With this program and the appropriate equipment you can grade tests for an entire class in just minutes. In addition to showing how to use Gradebook, the book explains the program's internal operation so that custom modifications can be made if desired.

In particular, the program:

- analyzes each test with high and low grades, the class average, and grade distribution
- summarizes how many questions were missed and what answers were chosen (when using optional card reader)
- provides for easy entry of grades from the keyboard
- allows you to curve the grades manually or automatically
- edits student data easily
- allows you to change your mind at any time about the number of tests and/or their relative weights

PRENTICE-HALL, INC., Englewood Cliffs, N.J. 07632

ISBN 0-13-362542-7